
Enero 1986 – 300 Ptas.
Todospectrum
Año II – Nº17
REVISTA EXCLUSIVA PARA USUARIOS
1986
Gráficos interactivos:
Dibujando la realidad
Basic:
Trucos
de programación
Mus:
Ordago a la grande
QL
Radiografía del QL
Gráficos con SuperBasic:
Colores

ASI ES EL QL, HECHO PARA NOSOTROS



SENCILLO

Para los profesionales que necesitamos un teclado en nuestro idioma, QL nos ofrece, en castellano, su QWERTY standar de 65 teclas móviles.
Para los que deseamos comunicarnos a gran velocidad y capacidad con nuestro ordenador, QL nos presenta su lenguaje SUPER BASIC.



ASEQUIBLE

Para los que necesitamos gran margen operativo, ahora disponemos de un ordenador con memoria ROM de 32K que contiene el sistema operativo QDOS, un sistema mono-usuario, multi-tarea y con partición de tiempo.



PROFESIONAL

Para los que deseamos tener perfectamente ordenada nuestra agenda de trabajo, presupuestos, fichas de productos, nuestra correspondencia, estadísticas de venta, archivo... QL viene dotado de cuatro microdrives totalmente interactivados entre sí: QL QUILL de Tratamiento de Textos, QL ARCHIVE Base de Datos, QL ABACUS Hoja Electrónica de Cálculo y el QL EASEL para realización de todo tipo de gráficos.



ALGUNAS DE LAS CONFIGURACIONES MAS USUALES:

QL	QL MONITOR MONOCROMO	QL MONITOR MONOCROMO IMPRESORA	QL MONITOR COLOR	QL MONITOR COLOR IMPRESORA
PVP 79.500 PTS	PVP 99.750 PTS	PVP 149.750 PTS	PVP 149.750 PTS	PVP 199.750 PTS



investronica

Tomás Bretón, 60. Telf. (91) 467 82 10. Telex 23399 IYCO E. 28045 Madrid
Camp. 80. Telf. (93) 211 26 58-211 27 54. 08022 Barcelona

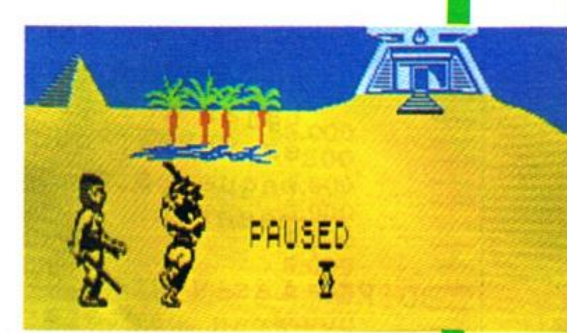
etc

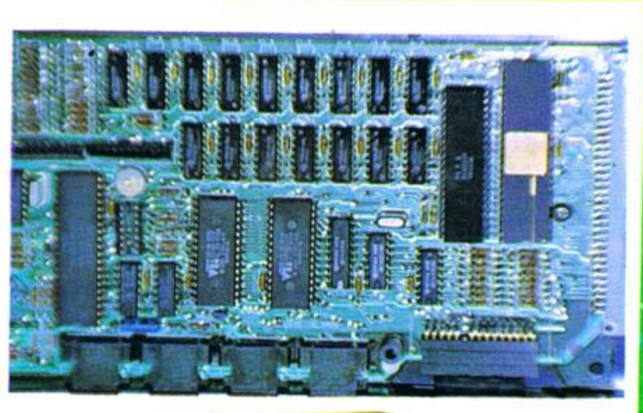


17

AÑO II







Ahora que ya estamos en las vacaciones de Navidad y a principio de un nuevo año llega el momento de realizar un paréntesis en el trabajo o en los estudios y dedicarse a nuestro querido ordenador. Para los amantes de los juegos de cartas ofrecemos un juego castizo, el mus. Pero esta no es la única novedad, una nueva serie dedicada a los gráficos interactivos hará las delicias de los programadores aficionados. También destacar el último artículo dedicado al nuevo sistema operativo que Manuel Arana ha diseñado. Esperamos que este primer número del año 1986 os guste y que la revista siga adelante con la colaboración de todos.

ACTUALIDAD

10 REM El Spectrum 128 ya está en el mercado. En la mayoria de los distribuidores de Investronica se puede encontrar ya el nuevo ordenador de Sinclair, pero no se puede decir lo mismo de los programas especificos para él; aunque esperamos que no tarden mucho.

20 REM Se avecina una autentica guerra de precios en el software: Por lo visto los fabricantes de programas no venden todo lo que quisieran y están decididos a bajar los precios todo lo que haga falta. Un ejemplo es SERMA, que a primeros de año lanza varios paquetes de programas al atractivo precio de 2.500 pesetas.

30 REM Las ventas del QL aumentan. Esta noticia se debe en gran parte a la espectacular bajada de precios que sufrió hace algunos meses. Esperemos que esto potencie las ventas de esta potente maquina.

40 REM Con la entrada de España en el Mercado Común los precios de ordenadores y otros productos informáticos van a subir. Sin embargo esta subida será menor para los productos fabricados en España. Aquellos que estén pensando en comprarse un Spectrum 128 están de enhorabuena.

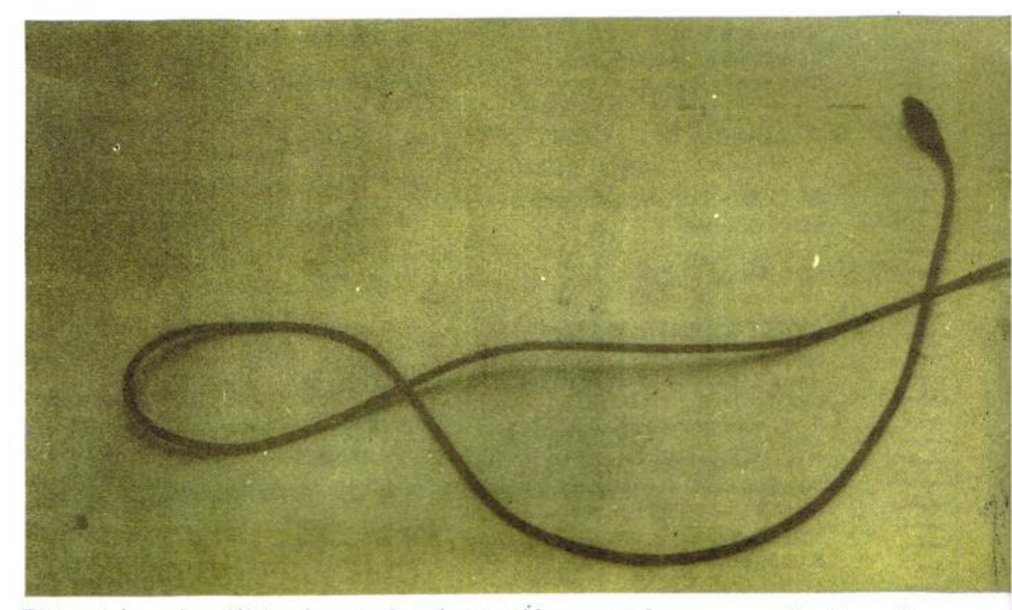
Gráficos interactivos

¿Estamos ante el arte del futuro?, es posible que así sea. El avance experimentado en el mundo de los gráficos por ordenador es impresionante, pero la metodología científica utilizada en su desarrollo logra que no sea un campo vedado exclusivamente a seres supersensibles, y que todo programador pueda obtener resultados brillantes, estéticamente hablando, incluso con el Spectrum.

Entendemos por Gráficos por Ordenador, o «Computer Graphics» a la disciplina o técnica que abarca la producción de imágenes gráficas con la ayuda de un ordenador. Esta definición es muy general y amplia, pero el campo de trabajo que permiten alcanzar los ordenadores en cuanto a materia de gráficos es prácticamente ilimitado, y la evolución que está experimentando esta técnica o arte es escalofriante. Uno de los inventos recientes más populares es el de los videojuegos, y un ejemplo de estos es el Ping Pong, que aunque ahora nos resulta muy simple, y por qué no decirlo, obsoleto, hace muy poco tiempo nos dejaba boquiabiertos. Los simuladores de vuelo, ya asequibles hasta en el Spectrum y con una gran calidad, hace una década hubieran resultado impensables.

¿Quién no se ha maravillado con las modernas películas de guerreros espaciales? Esas imágenes son gráficos producidos por un ordenador, que es capaz de simular todo tipo de luces, sombras, brillos, transparencias, explosiones, saltos al hiperespacio, planetas, en fin, realidades futuras que hemos empezado a vivir gracias a los estudios de una serie de programadores y matemáticos que la tecnología digital ha experimentado en estos últimos treinta años.

Otro ejemplo diario de aplicación de gráficos por ordenador lo tenemos en los titulares de televisión que se mueven y voltean sobre la pantalla y que también están controlados por un ordenador. Para nosotros estos gráficos son los pasivos, sólo podemos actuar como receptores, no tenemos acceso a modificarlos, son técnicamente llamados «Gráficos no Interactivos». En contraposición, llamamos «Gráficos Interactivos» a aquellos en los que los usuarios pueden controlar dinámicamente las imágenes a producir, el contenido, colores, etc.

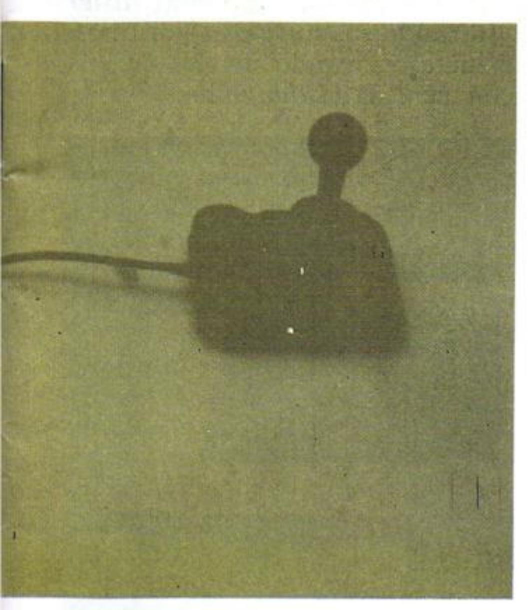


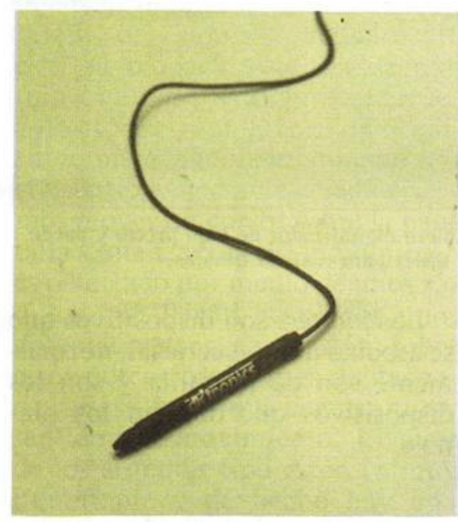
El *joystick* es el periférico de entrada más conocido, aunque no por sus aplicaciones de diseño gráfico, sino en los juegos.

Los gráficos Interactivos envuelven una doble vía de comunicación entre el usuario y el ordenador. El ordenador recibe señales desde un dispositivo de entrada y reacciona lo más rápidamente posible, instantáneamente para la percepción del usuario, esta respuesta llega al usuario a través del dispositivo de salida empleado generando o no una contrarrespuesta por parte del usuario que cierra el bucle. Este es el esquema generalizado más simple que explica la forma en la que se establece el diálogo entre programador o usuario con el ordenador.

La importancia de este campo no sólo está en los videojuegos. Los gráficos por ordenador tienen una aplicación importantísima en la industria. El diseño de circuitos electrónicos digitales, por ejemplo, es muy complejo y un ordenador puede facilitarlo enormemente, permitiendo incluso que puedan ser verificados por el ingeniero sobre el propio plano utilizando procedimientos de simulación. Los arquitectos pueden «visualizar» los resultados de sus diseños con perspectivas cambiantes de manera mucho más rápida. Los urbanistas y expertos en tráfico pueden simular situaciones y ver cómo se pueden solucionar. Los ordenadores nos pueden crear un mundo que se puede comportar como el nuestro y sobre el que podemos experimentar cualquier tipo de problema

por ordenador





Una alternativa barata del tablero digitalizador es el lápiz óptico. Aunque no permite resolución tan amplia.

o de solución y «ver» los efectos que producimos sin necesidad de crearlos realmente.

La historia

Pero, ¿cómo surgió todo esto? Como todo en esta vida, el desarrollo de estas técnicas ha requerido muchos años de investigación y estudio. En los años cincuenta se realizó el primer dispositivo gráfico controlado por ordenador, era un osciloscopio conectado al Whirlwind I del Massachusets Institute for Technologie (MIT) y fue utilizado para generar dibujos simples. Durante esta década los gráficos interactivos hicieron pequeños progresos porque los ordenadores de este período no eran utilizados para estos fines sino que se limitaban a ser «comedores de números» que permitían realizar cálculos a físicos y a ingenieros aeronáuticos para diseñar misiles. Al final de la década se desarrollaron los MIT TX 0 y TX 2 que hicieron factibles los gráficos interactivos con lo que el interés en este campo creció rápidamente.

El hecho que produjo el gran boom en este campo fue la tesis doctoral de Ivan E. Sutherland en 1962 titulada «Sketchpad: A man machine graphical communication system» que era un paquete de programas que probaron a muchos lectores que el campo de los gráficos interactivos era variable, utilizable y apasionante.

En la década de los sesenta se desarrollan grandes trabajos en este campo, en particular se desarrolló el CAD cuyas siglas significan «diseño asistido por ordenador», en empresas como General Motors, Bell Telephone, y Lockheed Aircraft, y comenzó la edad de oro que hoy disfrutamos.

Las utilizaciones actuales más importantes en gráficos interactivos son: representación de datos y resultados científicos, económicos, financieros, estadísticos... cartografía, animación y simulación, etc. En general cualquier programador que quiera realizar programas impactantes sobre el usuario debe recurrir al uso de gráficos interactivos, ya que ahora no basta con los ordenadores «comenúmeros».

El hardware

Los gráficos interactivos por ordenador requieren un soporte físico en el que realizarse, terminal de visualización, salidas en papel y entradas interactivas. A todos estos dispositivos se les denomina *hardware* y vamos a describir los dispositivos actualmente existentes parándonos especialmente en los que tenemos asequibles para el Spectrum.

En los sistemas de *hardware* podemos diferenciar entre las salidas gráficas y las entradas interactivas.

Dentro de las salidas gráficas, la más usual es el tubo de rayos catódicos CRT, en cualquiera de sus formas, raster, refresco, almacenamiento, etc. Un esquema clásico de un sistema gráfico interactivo basado en el CRT puede ser el siguiente. El ordenador va conectado a una unidad de proceso de visualización, también llamado DPU (Display Processing Unit) que es como otro ordenador especializado, con su propia memoria la mayor parte de las veces, que acepta una serie de comandos y ejecuta una serie secuencial de instrucciones, tales como trazar un segmento, dibujar un caracter... Hay dos formas en las que la DPU trabaja, el *«randon scan»* con el cual se accede a cada punto después de hacer un barrido línea a línea de toda la pantalla.

El modo *«random scan»* es también llamado vectorial y el *«raster scan»* es llamado simplemente *raster* y es el que implementa el Spectrum y por ello vamos a hablar un poco más de él.

Los terminales *raster* necesitan una memoria auxiliar donde almacenar los datos de la imagen, aunque en el Spectrum la memoria no es auxiliar ya que no se dispone de

Gráficos interactivos por ordenador

un «coprocesador» sino que es el mismo microprocesador Z80 el que se encarga de realizar el trabajo de la DPU. La memoria se codifica como una matriz de puntos en la que un cero significa blanco y un uno significa punto. La imagen se repite un número determinado de veces, que oscila entre 30 y 60 veces por segundo. Esta codificación puede ser rehecha para aumentar determinadas informaciones de brillo o color aunque en el Spectrum no se contempla esta posibilidad. Terminales *raster* de hasta 1.024 × 1.024 puntos son accesibles con relativa facilidad pero terminales de más puntos ya tienen problemas de refresco, es decir de falta de velocidad para que se mantenga la luminosidad del punto durante el tiempo que dura el barrido del resto de la pantalla. Por ahora este problema es crítico ya que si utilizamos pantallas de fósforos muy persistentes el borrado sería dificultoso.

Los terminales *raster* se abaratan de manera creciente, se permiten además colores, niveles de grises, etc., y se consiguen imágenes mucho más realísticas. El nivel de complejidad de la imagen no incrementa el parpadeo (lo que era el inconveniente más grave de los terminales vectoriales), permite borrados parciales, etc., por eso se han convertido en los terminales más usados en poquísimo tiempo y no resulta una frivolidad el decir que están de moda en esta década.

Las posibles alternativas CRT son los *«Plasma Panels»* en los que cada *pixel* consiste en una celdilla de gas con unos electrodos que pueden ser puestos a cero o a uno aplicando una tensión a los electrodos que no excedan o excedan a un determinado nivel crítico respectivamente. Estos terminales son de brillo atractivo, generalmente rojo y negro que permiten redes de alrededor de 1.000 × 1.000 puntos y que ya están en el mercado.

Los terminales de cristal líquido, comunes en los relojes digitales, pueden ser dispuestos en forma de matriz y constituir un terminal *raster pixel* por *bit*. Con diodos LED pueden obtenerse configuraciones similares.

Antes de abandonar estos tipos de sistemas gráficos con terminales *raster* debemos decir que próximamente veremos el funcionamiento del Spectrum como salida gráfica al estudiar el funcionamiento de la memoria gráfica, del ULA, etc.

Otro tipo de salidas son las no interactivas, que son las impresoras, los *plotters*, e incluso otras más sofisticadas que producen films e incluso hologramas.

Las impresoras fueron utilizadas en principio de modo gráfico para obtener dibujos de escasa calidad y precisión. Funcionan de modo similar al *raster* y con técnicas de sobreimpresión se pueden obtener tonos de gris. Actualmente son muy utilizadas para microordenadores definiendo los caracteres como matrices de puntos, estas impresoras son las llamadas «APA», All Point Adressible, es decir, con acceso a cualquier punto, que permiten su utilización como *raster* utilizando sus mismos algoritmos, e incluso creando niveles de gris con técnicas de *«halftoning»*.



El Koala Pad es una atractiva versión de un tablero digitalizador de bajo precio y puede considerarse como uno de los dispositivos más útiles para realizar dibujos.

Los *plotters* son dispositivos que se asocian a tipo vectorial, normalmente son de plumilla y son los dispositivos que dibujan los planos.

Dispositivos de entrada

Los dispositivos de entrada son los que permiten al usuario introducir las órdenes, relacionadas con la imagen, en el sistema. Hay muchos tipos de dispositivos que permiten esta introducción de órdenes por lo que sólo nos vamos a limitar a enumerarlos y dar una breve descripción de funcionamiento. Loca-

lizadores, permiten indicar una posición o una dirección sobre la imagen del terminal gráfico, entre ellos citaremos:

— Las tabletas digitalizadoras, que consisten básicamente en una superficie plana sobre la que se mueve un cierto apuntador, que puede ser una lupa con cursor o un lápiz, y a veces algunas teclas. Si el sistema está equipado con un lápiz, cuando este es apretado contra la superficie plana la localización de este punto es transmitida al ordenador marcando o borrando dicho *pixel.* Si el sistema mueve una lupa con cursor se suelen utilizar las teclas para transmitir su posición. En aplicaciones interactivas gráficas, es normal el transponer la superficie plana a la pantalla y representar como un aspa la posición del cursor sore la visualización gráfica sobre la que estamos trabajando. Se emplea comúnmente en cartografía, medicina, etc. El ratón, o *«mouse»,* actualmente está de moda y sistemas de este tipo están empezando a aparecer para el Spectrum. Este es un dispositivo manejado manualmente por el usuario y mediante el cual podemos mover un cursor sobre la pantalla visualizadora. El movimiento es detectado por medio de unos rodillos que llevan acoplados unos potenciómetros, la tensión sobre estos potenciómetros es digitalizada y así se determina la magnitud y sentido del movimiento. El auge de los sistemas tipo ratón es muy importante y de hecho hay una empresa que ha lanzado su último ordenador recientemente y que la campaña de publicidad que han realizado ha consistido en decir que lleva incorporado un ratón (el Apple Macintosh).

— Los *«trackballs»* son unos dispositivos que consisten en una esfera que el usuario puede hacer girar sobre su centro en cualquier dirección. Al girar la esfera mediante unos potenciómetros y de manera similar al ratón detecta los movimientos del cursor y los reproduce sobre la pantalla. Este sistema está cayendo en desuso ya que aunque sea muy preciso para pequeños movimientos es muy lento para grandes desplazamientos, por ello los *joysticks* están ocupando este espacio.

— Los *«joysticks»* que son simplemente una palanca que puede moverse según cuatro u ocho direcciones según el modelo. El movimiento puede ser detectado mediante unos potenciómetros de manera similar al ratón o de cualquier otra manera. Hay dos tipos de *joysticks,* los de posición absoluta y los de posición incremental. El de posición absoluta coloca el cursor a lo alto de la pantalla cuando se mueve hacia adelante, y lo sitúa en la parte inferior de la pantalla cuando se mueve hacia atrás. El posicionamiento es muy rápido y por consiguiente muy difícil de apuntar. El *joystick* incremental es como su propio nombre indica de funcionamiento incremental, es decir, que a partir de la posición que ocupa dependiendo de hacia dónde desplacemos la palanca moveremos el cursor. Actualmente ha surgido un *joystick* híbrido que mediante una tecla puede funcionar o como absoluto o como incremental conjugándose así las ventajas. Para los Spectrum tenemos bastante variedad de *joysticks* en el mercado.

Los gráficos interactivos por ordenador requieren un soporte físico (Hardware)

— Otro tipo de entrada interactiva muy sofisticado es la Tableta Digitalizadora Tridimensional cuyo funcionamiento es idéntico al de la bidimensional pero en lugar de detectar la posición mediante tensiones sobre potenciómetros utiliza ondas sónicas.

— Los «selectores» son dispositivos con los que, mediante un lápiz luminoso podemos seleccionar una opción entre varias que se muestran en la pantalla. Estos sistemas están pensados para que el diálogo interactivo sea más sencillo y no se requiera personal especializado para realizar un trabajo más sencillo.

— Los sistemas de «Elección Numérica» son aquellos en los que seleccionamos un número entero positivo y se usa para elegir entre varias opciones en las que cada una tiene asignado un número. El sistema que permite la elección puede ser cualquiera aunque lo más normal es utilizar el teclado. El teclado es utilizado cuando queremos que la entrada sea una «cadena de caracteres». Un ejemplo muy sencillo de este tipo de entrada es cuando durante la ejecución del programa se nos presenta si queremos hacer algo y nos piden que contestemos «Sí o No».

— Lo más impresionante dentro del campo de las entradas interactivas lo constituyen los reconocedores de voz, que aunque hoy estén un poco alejados de la utilización práctica constituyen una posibilidad futura muy a tener en cuenta.

FIGHTING WARRIOR

Distribuidor: Erbe
Memoria requerida: 48 K
Precio: 2.100 ptas.

Trasladados al antiguo Egipto nos encontramos con la maldad de un Faraón que haciendo uso del gran poder de que dispone ha secuestrado a una pricesa joven e indefensa. Ante esta situación nos ofrecemos voluntarios para ir a rescatarla, sin más defensa que una espada y ciertos objetos abandonados en el desierto con poderes mágicos, pero que no siempre nos harán bien y que debemos averiguar cómo utilizarlos.

Los peligros más importantes que se nos presentan son unos animales mitológicos, mitad hombres mitad dioses, que nos supondrán un serio obstáculo, impidiéndonos el paso. Además de estos tendremos que tener cuidado con unas enigmáticas flechas que debemos evitar si no queremos ver mermadas nuestras fuerzas. Nosotros disponemos de una fuerza marcada en la parte inferior de la pantalla, debajo de nuestro guerrero, que vamos perdiendo a medida que recibimos los golpes, aunque después de cada combate recuperamos un poco, dependiendo del tiempo que tardemos en derribar al contrario. El punto de partida es el desierto, que tendremos que atravesar si queremos llegar al templo donde se encuentra la princesa y donde nuestros problemas aumentarán de forma considerable.

Los guerreros son de un gran tamaño lo que permite que su definición gráfica sea bastante buena y que tengan unos movimientos claros y atractivos, aunque con el inconveniente de que son un poco lentos y resulta difícil controlar el tiempo en el que van a reaccionar. El juego encierra una gran dificultad, resultando más difícil derribar contrarios cuanto más avances en él, aunque al utilizar los objetos abandonados en el camino correctamente, te será mucho más fácil alcanzar el objetivo.



Nosotros manejamos el luchador situado a la izquierda y debemos matar a todos los enemigos que nos surjan.



Una situación no demasiado extraña: Los dos oponentes mueren simultáneamente.

Control: Teclado, cursor, Joystick.

Jugadores: Uno.

Gráficos: Notables aunque el movimiento no es todo lo rápido que uno quisiera.

Nivel de dificultad: Aumenta según los contrarios derribados, y la proximidad al objetivo.

Originalidad: La originalidad del juego reside básicamente en la manera que tienen los guerreros de luchar, que es de un realismo espectacular.

Conclusión: Es este un juego emocionante que no resta interés hasta que consigues el objetivo, cosa que no te será nada fácil sin alguna práctica.

FAIRLIGHT

Distribuidor: ABC
Memoria requerida: 48 K.
Precio: 2.300 ptas.

A primera vista te parecerá un juego como todos, pero éste difiere de los demás en que no te marca el objetivo a seguir, eres tú quien recorriendo e investigando en la dependencias del castillo debes descubrir su misterio. Quizás esto te lleve mucho más tiempo del previsto pero no te desanimes, valdrá la pena buscar.

El juego está realizado en tres dimensiones y fue diseñado con una de las más avanzadas técnicas de programación desarrolladas hasta el momento, que le da gran realismo y claridad a los gráficos y movimientos de nuestro héroe.

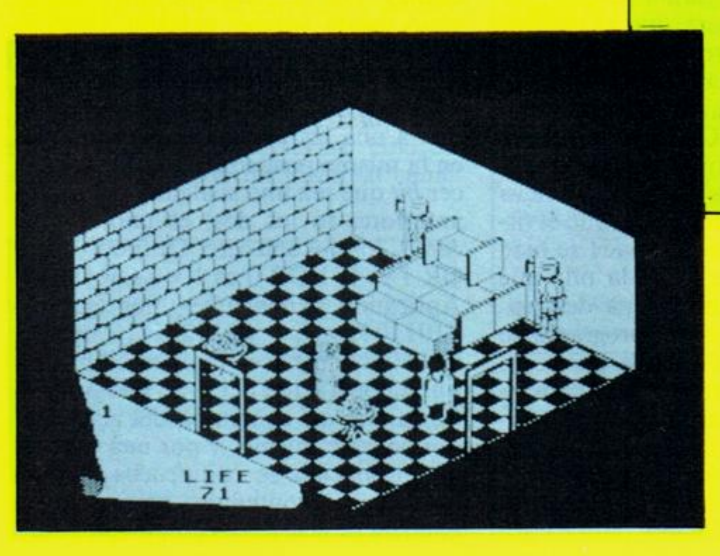
El juego se desarrolla en el interior de un castillo de la Edad Media, cuya salas o lugares estratégicos están defendidos por guardianes armados, gigantes o fantasmas que tratarán de proteger el misterio, combatiendo contigo si intentas pasar.

Esparcidos por todo el castillo se encuentras objetos dispares como llaves, comida, sacas de dinero, relojes de arena, etc., que podrás guardar en cada uno de los cinco bolsillos que tienes o utilizarlos si lo consideras preciso.

El sistema de vida que utiliza es distinto a los demás juegos conocidos; en este dispones de 99 vidas que irás perdiendo con bastante rapidez cuando seas atacado. Aunque también podrás recuperarlas utilizando algunos de los objetos que encuentres, todos ellos tienen una utilidad específica que deberás averiguar, aunque algunos no son lo que aparentan.

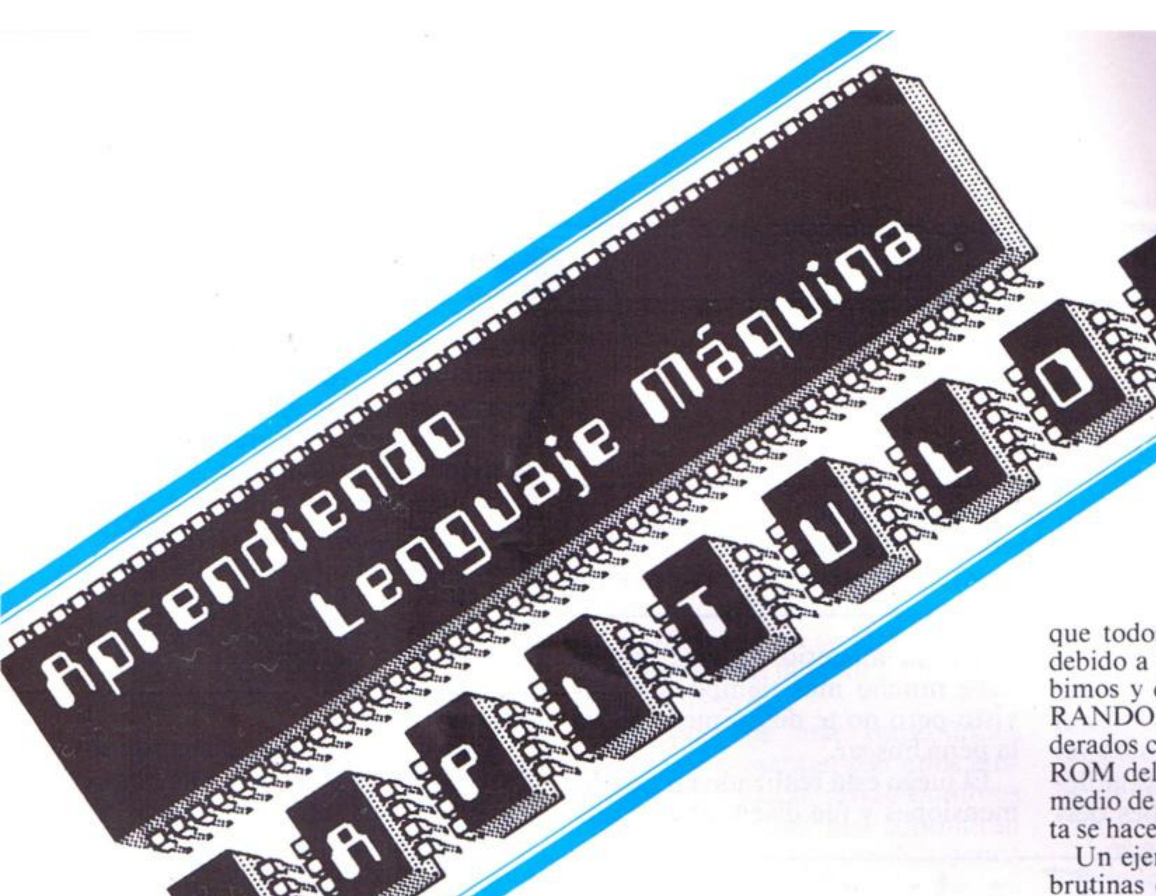


La pantalla de presentación del juego es una auténtica maravilla de detalles y colores.



La sala del trono, en la que nuestro héroe se enfrenta a un enemigo al que no puede derrotar.

Control: Teclado o joystick.
Jugadores: Uno.
Gráficos: Diseñados con una avanzada técnica en 3-D, con una buena definición y colorido muy vistoso.
Nivel de dificultad: Bastante elevado.
Originalidad: Muy original. Quizás este programa abra paso a una nueva etapa en la estructura de los videojuegos.
Conclusión: En este juego no es sólo imprescindible la habilidad de manejo del joystick o teclado, si no también es necesaria la habilidad intelectual.



Ya hemos visto cómo sumar y restar en código máquina utilizando múltiple precisión. Los programas que hacían esto eran relativamente pequeños y no ofrecían dificultad. El único problema es que si quisiéramos operar con una precisión superior a dos *bytes*, tendríamos que hacer crecer el programa proporcionalmente, y en el caso de una suma o resta de 40 *bytes*, el programa adquiriría proporciones descomunales.

Pero nos podemos evitar este problema usando subrutinas. Estas estructuras son similares a las subrutinas del BASIC y su manejo es básicamente el mismo pero a nivel de lenguaje máquina. Su uso nos permite que determinadas partes de un programa que se repiten muy a menudo puedan ser escritas una sola vez y llamadas desde distintas partes del código principal.

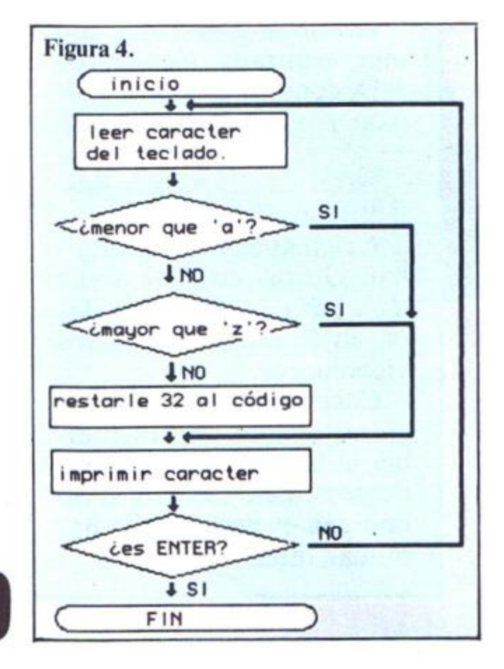
La instrucción que se utiliza para llamar a una subrutina (el GOSUB del BASIC) es CALL XXXX, donde XXXX es la dirección de dieciséis *bits* de la rutina (o su etiqueta correspondiente). En dicha dirección debe encontrarse el inicio del bloque de instrucciones que queremos usar como subrutina. Al final de dicho bloque, en el lugar en que se encontraría el RETURN en un programa BASIC, se debe poner RET para que el control vuelva a la siguiente instrucción a la que realizó la llamada. Para aquellos que trabajen directamente en hexadecimal, diremos que CALL es "CD" seguido de los dos *bytes* de la instrucción, y que RET es "C9" y no lleva ningún parámetro detrás, ya que el ordenador recuerda desde dónde se hizo la llamada (por medio de la pila que veremos más adelante). Una descripción gráfica del flujo del programa se muestra en la figura 1, en la que se ve cómo desde una instrucción se llama a una rutina situada al final del programa y al terminar esta el control se devuelve a la instrucción siguiente.

Si se ha fijado en los pequeños listados que hemos publicado, habrá visto que todos terminan en RET. Ello es debido a que los programas que escribimos y que llamamos por medio del RANDOMIZE USR xxxx, son considerados como subrutinas por la propia ROM del Spectrum, que los llama por medio de un CALL y por tanto la vuelta se hace con RET.

Un ejemplo de aplicación de las subrutinas es el listado que publicamos en la figura 1. Este realiza la misma suma de dieciséis *bits* que vimos en el capítulo anterior pero usando subrutinas, de modo que las sumas *byte* a *byte* se hacen siempre en el mismo sitio.

El proceso detallado se puede dividir en dos bloques. En primer lugar veamos la subrutina. Esta coge el *byte* al que apunta IX y le suma el contenido de IX+2 más el acarreo; almacenando el resultado en IX+4 y devolviendo el control por medio de un RET. Con esta subrutina el programa principal es muy sencillo. En primer lugar cargamos en IX el valor 60501, que es la parte inferior del primer número a sumar. A continuación ponemos el acarreo a cero. Esto se hace porque según indicamos, el *byte* más inferior se sumaba sin acarreo, pero eso era lo mismo que sumar con acarreo 0, que es lo que tenemos que hacer en este caso ya que la subrutina suma siempre con acarreo. La instrucción usada para esto es "OR A". Aunque esto en principio pueda parecer extraño, tiene su razón de ser y es uno de los trucos más usados del código máquina del Z-80. El OR de dos *bytes* lo que hace es coger una a una las parejas de *bits* situadas en la misma posición y generar un tercer *bit* que sea uno si alguno de los dos anteriores (o los dos) es uno, y vale cero si los dos son cero. De modo que 1 OR 1 es 1, 1 OR 0 es 1 y 0 OR 0 es 0. Aplicando a dos *bytes*, por ejemplo 01011011 y 00110111, queda: 01111111 al hacer la operación *bit* a *bit*. La operación del Z-80 OR A realiza un OR del contenido del acumulador consigo mismo, y por una sencilla comprobación se comprueba que no modifica el contenido, pero pone la bandera de acarreo a cero, lo que resul-

Figura 4.



ta de gran interés para nuestros propósitos.

Una vez inicializados estos dos parámetros, llamamos a la subrutina, que sumará automáticamente el primer *byte* de cada número y guardará el resultado en sitio correspondiente. A continuación decrementamos IX por medio de la instrucción DEC IX, similar en funcionamiento a INC HL, pero que en lugar de incrementar ese registro de 16 *bits*, decrementa el IX (también de 16 *bits*) en una unidad. Su equivalente hexadecimal es "DD 2B". Al realizar esta acción, IX se queda con un valor de 60500, que corresponde al segundo *byte* de la suma, operación que se realiza inmediatamente al llamar de nuevo a la subrutina. En este caso el acarreo es el resultado de la suma anterior y no hay que tocarlo, labor que se facilita al no ser modificado por ninguna instrucción de las que usamos entre llamadas a la subrutina.

Sumas con más bytes

Esta idea que hemos dado para las sumas de números de dos *bytes* se puede ampliar a números mayores, para lo que bastaría en principio con repetir las instrucciones de decrementado de IX y de llamado a la subrutina. Esto se hace en el listado de la figura 2, que suma dos números de 3 *bytes*. Para sumar más *bytes* basta añadir las instrucciones DEC IX y CALL a la subrutina, modificando, eso sí, las direcciones de memoria donde se almacenan los datos para que no es escriban unos encima de otros.

La máquina de escribir

En todos los programas que hemos visto hasta ahora las entradas y salidas las hacíamos desde BASIC. Pero también es posible hacerlo desde lenguaje máquina utilizando para ello la ROM del BASIC que ya trae un conjunto de rutinas con utilidades varias. Una de ellas se encarga de imprimir al canal

Figura 1.

```
EA60               0010        ORG  60000
EA60 DD2155EC      0020        LD   IX,60501
EA64 B7            0030        OR   A
EA65 CD6EEA        0040        CALL SUMAB
EA68 DD2B          0050        DEC  IX
EA6A CD6EEA        0060        CALL SUMAB
EA6D C9            0070        RET
EA6E DD7E00        0080 SUMAB  LD   A,(IX+0)
EA71 DD8E02        0090        ADC  (IX+2)
EA74 DD7704        0100        LD   (IX+4),A
EA77 C9            0110        RET
                   0120        END
SUMAB EA6E    #    5DCE
```

que se encuentre activo en ese momento (normalmente la pantalla). Se llama por medio de RST 16 (10h) e imprime el caracter que haya en el acumulador. La instrucción RST es una versión rápida del CALL que se utiliza para determinadas direcciones bajas de memoria (las que son múltiplos de 8) y su código hexadecimal varía con cada dirección, ya que en un solo *byte* se mete la dirección y la dirección de llamada.

Otra llamada que también resulta muy interesante es la de leer un caracter del teclado, CALL 5598, que devuelve en A el caracter leído. Con ello podemos hacer un sencillo programa de máquina de escribir que se limite a copiar en pantalla lo que se teclee, como se demuestra en el listado de la figura 3.

Por último queda una nueva instrucción que es necesaria para hacer el programa, esta es el salto incondicional, similar al GOTO del BASIC, que nos permite que después de leer e imprimir un caracter, vuelva al principio para leer otro. Se escribe JP xxxx y xxxx es una dirección de 16 *bits*, la función que hace es similar al CALL pero con la excepción de que no guarda en la pila la dirección desde la que se hizo la llamada y por tanto no se puede volver con un RET. Esto es similar a la diferencia entre GOSUB y GOTO en BASIC.

Una vez realizado este sencillo programa, podemos hacer que la máquina sea algo más inteligente y que sólo escriba en mayúsculas, modificando las minúsculas. La estructura básica es la misma del listado anterior, pero añadiendo instrucciones después de la lectura del teclado y antes de la impresión en pantalla, de modo que si es una minúscula, es decir, si su código se encuentra entre el 97 y el 122 (la tabla

Figura 2.

```
EA60               0010        ORG  60000
EA60 DD2156EC      0020        LD   IX,60502
EA64 B7            0030        OR   A
EA65 CD73EA        0040        CALL SUMAB
EA68 DD2B          0050        DEC  IX
EA6A CD73EA        0060        CALL SUMAB
EA6D DD2B          0070        DEC  IX
EA6F CD73EA        0080        CALL SUMAB
EA72 C9            0090        RET
EA73 DD7E00        0100 SUMAB  LD   A,(IX+0)
EA76 DD8E03        0110        ADC  (IX+3)
EA79 DD7706        0120        LD   (IX+6),A
EA7C C9            0130        RET
                   0140        END
SUMAB EA73    #    5DCE
```

Figura 4.

```
EA60               0010        ORG  60000
EA60 CDDE15        0020 LEE    CALL 5598
EA63 D7            0030        RST  16
EA64 C360EA        0040        JP   LEE
                   0050.       END
LEE   EA60    #    5DCE
```

completa de caracteres del Spectrum se encuentra en el apéndice A del manual de programación), hay que restarle 32 para que se convierta en su equivalente en mayúsculas. Otra innovación que vamos a introducir es que al pulsar ENTER (que tiene el código ASCII 13), se termine el programa y volvamos al BASIC.

El organigrama del programa se muestra en la figura 4 y para realizar las correspondientes modificaciones utilizaremos unas nuevas instrucciones, las de comparación y salto condicional.

La instrucción de comparación, CP, se realiza entre el acumulador y el dato que se indica detrás. Lo que hace es una resta de dicho contenido menos el dato indicado a continuación pero sin alterar el contenido del acumulador, por lo que lo único que resulta afectado son las banderas. En nuestro caso tenemos que comprobar primero si el código del caracter leído es menor que "a" (97), para ello comparamos, por lo que al realizar la seudo-resta la bandera de acarreo se pondrá alta si el resultado es negativo, es decir, si el caracter leído es mayor. Para la segunda comparación (si es mayor que "z") utilizamos 123, que es el código siguiente al de ese caracter.

Por último quedan los saltos condicionales. Estas instrucciones son del tipo del JP, pero con dos importantes diferencias. La primera es que el salto sólo se realiza si se cumple la condición indicada, en el caso de JR C,xx, si la bandera de acarreo está a uno; mientras que en JR NC,xx el salto se realiza si el acarreo está bajado. La otra diferencia es que la dirección ocupa un *byte* tomado como un número en complemento a dos, por lo que sólo puede direccionarse un rango de direcciones comprendido entre 127 y –128 respecto al inicio de la siguiente dirección, que es la que se toma como 0. Cuando se trabaja en ensamblador no hace falta ir contando, ya que al poner la etiqueta el programa calcula automáticamente el desplazamiento necesario.

Con todo esto se realiza un nuevo programa que se lista en la figura 5.

Figura 5.

```
EA60               0010        ORG  60000
EA60 CDDE15        0020 LEE    CALL 5598
EA63 47            0025        LD   B,A
EA64 FE61          0030        CP   "a"
EA66 3806          0040        JR   C,IMPRI
EA68 FE7B          0050        CP   123
EA6A 3002          0060        JR   NC,IMPRI
EA6C D620          0070        SUB  32
EA6E D7            0080 IMPRI  RST  16
EA6F 78            0085        LD   A,B
EA70 FE0D          0090        CP   13
EA72 20EC          0100        JR   NZ,LEE
EA74 C9            0110        RET
                   0120        END
IMPRI EA6E   LEE   EA60   #    5DDB
```



Un operativo para el Spectrum

AUTO

La definición de este comando es muy sencilla, ya que toda su sintaxis se ha especificado en las tablas, y su ejecución consiste únicamente en pasarle los parámetros al bucle principal, que ya se encargará de escribir los números de línea. Lo primero que hace es recuperar los parámetros del número de línea e incremento de la pila de cálculo y los resta para obtener el valor correcto de la variable AUTOLI. Por último, inicializa las variables AUTOIN y AUTON y acaba saltando directamente sobre el bucle principal.

MODE

Al igual que con AUTO la sintaxis está definida en las tablas. Durante la ejecución coge el número de la pila de cálculo, comprueba que está dentro de los límites correctos y busca el valor correspondiente para la variable PRDIR (dirección de la rutina de salida) en la tabla TPRO.

ON ERR

Este es un comando de clase 5, y por lo tanto, todo el trabajo de análisis sintáctico debe hacerlo la propia rutina ONERR. Como compensación la ejecución es muy simple, ya que se comporta como si se tratara de un REM. (Sólo se pasa detrás cuando se ha producido un error). Para realizar el análisis sintáctico utiliza la rutina LIERR, que se encarga de comprobar los parámetros de los códigos de error. Esta rutina también se utiliza cuando se ha producido un error para comprobar si hay alguna referencia al error producido en una sentencia ONERR. En este caso nos devolverá el indicador de cero a uno si se ha encontrado el código.

ELSE

El peso del comando ELSE está dentro de la rutina de ejecución de un IF. La rutina de ELSE se reduce a continuar si se está realizando un análisis sintáctico o a pasar a la siguiente línea si estamos en ejecución. De esta forma, la rutina queda reducida a solamente tres instrucciones. Lo que queda detrás de ELSE sólo se ejecutará si no se cumple la condición del IF correspondiente.

SWAP

Si se llama a este comando durante el análisis sintáctico la rutina comprueba la existencia de los dos nombres de variables separados por una coma llamando dos veces a 1CIF (CLASS 01), y luego comprueba que las dos variables son del mismo tipo.

Si se llama durante la ejecución el tratamiento es distinto, según si se trata de un cambio de variables numéricas o alfanuméricas. Si son numéricas simplemente coge los dos valores en la pila de cálculo y luego los asigna en orden inverso llamando dos veces a LET. Si son variables alfanuméricas no es posible hacer esto, ya que lo que se almacena en la pila no es el propio valor, sino una serie de parámetros que nos indican dónde está la cadena y qué longitud tiene, y si lo intentáramos destruiríamos una de las variables al hacer la primera asignación. Por esta razón, una de las variables debe ser copiada en el área de trabajo antes de realizar las dos transferencias.

ON

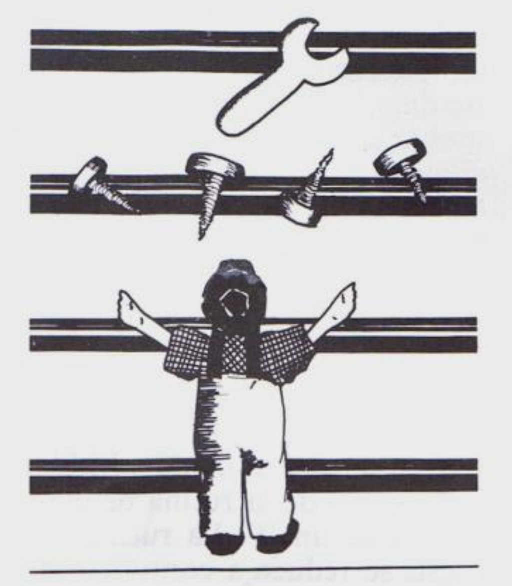
Este comando realiza de una forma paralela el análisis sintáctico y la ejecución, aprovechándose de que la mayoría de las rutinas de la ROM que necesita están preparadas para ello. En primer lugar, coge el valor de la expresión (o comprueba su sintaxis) y mira si detrás hay un GO TO o GO SUB; a continuación, entra en un bucle en el que va cogiendo todos los valores de los números de línea y luego los saca de la pila a no ser que sea el que nos interesa. Una vez acabado este proceso da por terminada la rutina si está realizando un análisis sintáctico, o pasa a la rutina GOTO o GOSUB si está ejecutando.

RENUM

Este es sin duda el más complejo de los nuevos comandos. Cuando este comando es llamado los dos primeros parámetros de la sentencia (que siempre existen) ya han sido introducidos en la pila (o analizados). Lo primero que debe hacer es comprobar si existen los otros dos y analizarlos si es así. Una vez hecho esto, hemos acabado en el caso de análisis sintáctico. Si estamos en ejecución hay que colocar los valores por defecto si es necesario y comprobar si todos los parámetros son correctos. El número de la última línea a numerar debe ser mayor que el de la primera línea, el incremento no debe ser cero, y el

Este mes acabamos la serie de artículos dedicados al nuevo sistema operativo con la parte correspondiente a las rutinas de los siete nuevos comandos y el modo de 64 caracteres. Con todas las rutinas de los números anteriores ya está todo preparado para que cada comando sea llamado en su momento y, que éste pueda apoyarse sin problemas en la ROM y en el resto del programa, con lo que la definición del comando se ha simplificado al máximo.

nuevo número de la última línea que se va a renumerar debe ser menor que el número de la siguiente línea. Además de éstas se deberían haber hecho otras comprobaciones, como que el incremento no sea negativo. Durante la ejecución de RENUM se utiliza en varias ocasiones la rutina NNL, a la cual se le pasa el número de línea en los registros BC y nos devuelve en IX el número de línea que le corresponderá después de la renumeración. Además, nos devolverá el indicador de *carry* a uno si la línea no va a ser modificada. Esta rutina se utiliza por primera vez para



calcular el nuevo número que le corresponde a la última línea a renumerar. A la vez que se realizan estas comprobaciones se inicializan las variables del sistema con los cuatro parámetros, para que puedan ser usados por NNL. Una vez hecho esto, se llama a la rutina SCAN que rastrea todo el programa en busca de comandos que pueden necesitar ser modificados, y los arregla si es necesario. Por último, cambia los números de las líneas indicadas. Para ello marca el final de la zona a renumerar como si fuera el final del programa durante el proceso, y lo vuelve a arreglar al final.

NEW

La rutina NEW nos permite borrar el programa BASIC e

```
AUTO    CALL  ^1E99
        PUSH  BC
        CALL  ^1E99
        POP   HL
        LD    D,B
        LD    E,C
        EX    DE,HL
        AND   A
        SBC   HL,DE
        LD    (AUTOLI),HL
        LD    (AUTOIN),DE
        LD    HL,AUTON
        LD    (HL),^01
        LD    SP,(^5C3D)
        POP   AF
        JP    MAIN1
MODE    CALL  ^2DD5
        JR    C,REPB
        CP    ^02
        JR    C,MOD1
REPB    RST   8
        DEFB  ^0A
MOD1    ADD   A,A
        LD    C,A
        LD    B,^00
        LD    HL,TPRO
        ADD   HL,BC
        LD    E,(HL)
        INC   HL
        LD    D,(HL)
        LD    (PRDIR),DE
        RET
TPRO    DEFW  ^09F4
        DEFW  PROUT1
ONERR   CALL  ^2530
        JP    NZ,REM
        CALL  LIERR
GSTL1   CALL  GETCH
        JP    STMTL1
LIERR   CALL  GETCH
        CALL  ^2C88
        JR    NC,REPB
        PUSH  AF
        EX    AF,AF'
        POP   AF
ONER1   CP    ^3A
        JR    C,LETRA
        AND   ^DF
        SUB   ^07
LETRA   SUB   ^30
        CP    E
        JR    NZ,ONER2
        EX    AF,AF'
ONER2   CALL  NEXTCH
        CP    ^2C
        JR    NZ,ONERET
        CALL  NEXTCH
        CALL  ^2C88
        JR    C,ONER1
        RST   8
        DEFB  ^0B
ONERET  EX    AF,AF'
        RET
ELSE    CALL  ^2530
        JP    NZ,REM
        JR    GSTL1
SWAP    CALL  ^2530
        JR    NZ,SWRUN
```

inicializar el sistema sin desconectarnos del resto del programa. Este NEW no vuelve a testear toda la memoria disponible, sino que se limita a inicializar las variables indicando que no existe programa BASIC.

COMND

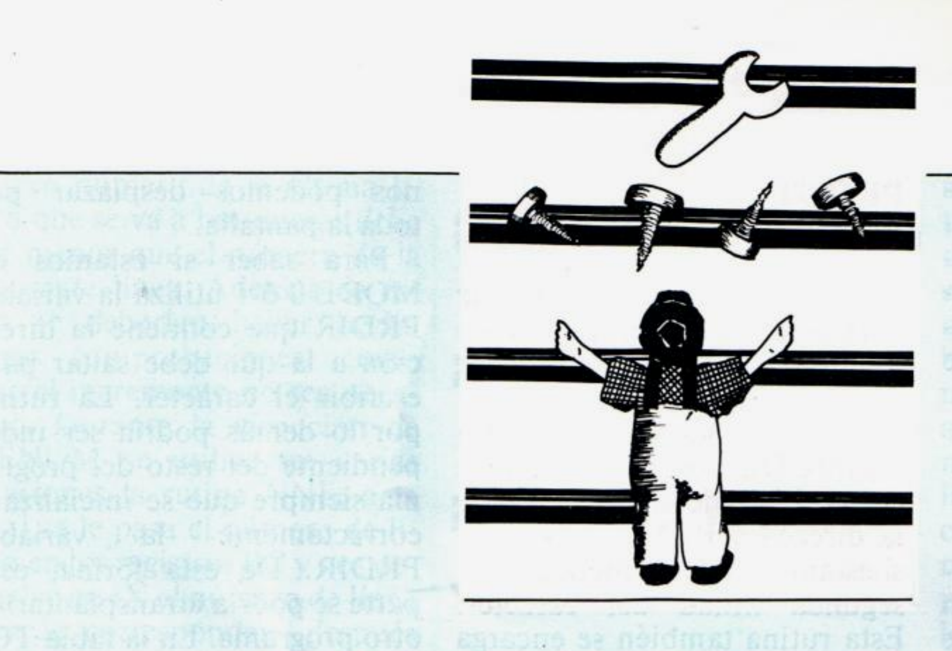
Esta es la rutina de los comandos que todavía no han sido definidos. Lo único que hace es generar un mensaje de error. Este es el punto en que se debería escribir el hipotético octavo comando correspondiente al código 31.

PROUT

Es la encargada de gestionar toda la impresión en pantalla e impresora, tanto en modo de 32 como de 64 caracteres. Si estamos en el modo de 64 caracteres (MODE 1) el programa modifica el *byte* de salto relativo en la dirección PAMOD+1 según si estamos en la primera o en la segunda mitad del caracter. Esta rutina también se encarga del tratamiento de los caracteres de control. En este punto se han corregido los errores que se producían al escribir los caracteres 8 y 9, con los cuales ahora nos podemos desplazar por toda la pantalla.

Para saber si estamos en MODE 0 ó 1 utiliza la variable PRDIR que contiene la dirección a la que debe saltar para escribir el caracter. La rutina por lo demás podría ser independiente del resto del programa siempre que se inicializara correctamente la variable PRDIR. De esta forma, esta parte se podría «transplantar» a otro programa. En la tabla TOTAB están los nombres de las palabras clave de los nuevos comandos con el *bit* más significativo del último *byte* puesto a uno.

```
        CALL ^1C1F
        LD   C,^2C
        CALL SEPARA
        LD   A,(^5C3B)
        PUSH AF
        CALL ^1C1F
        POP  AF
        LD   D,(IY+1)
        XOR  D
        AND  ^40
        RET  Z
REPC2   RST  8
        DEFB ^0B
SWRUN   LD   HL,(^5C5D)
        PUSH HL
        CALL ^24FB
        BIT  6,(IY+1)
        CALL Z,SWSTR
        CALL NEXTCH
        CALL ^24FB
        POP  HL
        LD   (^5C5D),HL
        CALL ^1C1F
        CALL ^2AFF
SWCON   CALL NEXTCH
        CALL ^1C1F
        CALL ^2AFF
        RET
SWSTR   CALL ^2BF1
        LD   A,B
        OR   C
        JP   Z,^2AB2
        POP  HL
        PUSH DE
        RST  ^30
        POP  DE
        PUSH BC
        LD   HL,(^5C61)
        EX   DE,HL
        LDIR
        CALL NEXTCH
        CALL ^24FB
        POP  BC
        POP  HL
        PUSH BC
        LD   (^5C5D),HL
        CALL ^1C1F
        CALL ^2AFF
        LD   DE,(^5C61)
        POP  BC
        CALL ^2AB2
        JP   SWCON
ONAGO   CALL GETCH
        CALL ^1C82
        LD   B,A
        AND  ^FE
        CP   ^EC
        JR   NZ,REPC2
        LD   C,^00
        PUSH BC
        CALL ^2530
        SCF
        CALL NZ,^2DD5
        JR   NC,ONLOOP
        LD   A,^FF
ONLOOP  PUSH AF
        CALL NEXTCH
        CALL ^1C82
```



```
        POP   AF
        CALL  ^2530
        JR    Z,ON1
        DEC   A
        JR    Z,ON2
        PUSH  AF
        RST   ^28
        DEFB  ^02,^38
        POP   AF
ON1     LD    B,A
        CALL  GETCH
        CP    ^2C
        LD    A,B
        JR    Z,ONLOOP
        POP   AF
        CALL  ^2530
        RET   Z
        RET   NC
        CP    ^ED
        JP    Z,^1EED
        JP    ^1E67
ON2     POP   BC
        DEC   C
        PUSH  BC
        JR    ON1
RENUM   CALL  ^2530
        JR    NZ,RERUN
        RST   ^18
        CP    ^2C
        RET   NZ
        RST   ^20
        JP    ^1C7A
RERUN   RST   ^18
        CP    ^2C
        JR    Z,CUATRO
        LD    HL,^2710
        PUSH  HL
        LD    BC,^0000
        JR    DOSPAR
CUATRO  RST   ^20
        CALL  ^1C7A
        CALL  ^1E99
        PUSH  BC
        CALL  ^1E99
DOSPAR  POP   HL
        AND   A
        SBC   HL,BC
        ADD   HL,BC
        JR    C,REPB2
        LD    (ULTLI),HL
        LD    (PRILI),BC
        CALL  ^1E99
        LD    A,B
        OR    C
        JR    NZ,REN1
REPB2   RST   8
        DEFB  ^0A
REN1    LD    (INCR),BC
        CALL  ^1E99
        LD    (COMIE),BC
        LD    BC,(ULTLI)
        CALL  NNL
        PUSH  IX
        CALL  Z,^19B8
        LD    D,(HL)
        INC   HL
        LD    E,(HL)
        POP   HL
        AND   A
        SBC   HL,DE
        JR    NC,REPB2
        LD    HL,(^5C5D)
        LD    (^5C5F),HL
        CALL  SCAN
        LD    A,^FE
        CALL  ^1601
        LD    HL,(^5C5F)
        LD    (^5C5D),HL
        LD    (IY+38),^00
        LD    HL,(ULTLI)
```

Todospectrum



TODOSPECTRUM es una publicación mensual que le ayudará a obtener el máximo partido a su **SPECTRUM** y al **ZX 81**.

Sensacional Oferta de Suscripción



CONOZCA LAS VENTAJAS DE SUSCRIBIRSE A

Todospectrum



ADEMAS, le hacemos un 25 % DE DESCUENTO

sobre el precio real de suscripción (12 números)

VALOR REAL DE SUSCRIPCION	OFERTA ESPECIAL DE SUSCRIPCION	USTED AHORRA
~~3.600~~ PTAS.	2.700 PTAS.	900 PTAS.

APROVECHE AHORA esta oportunidad irrepetible para suscribirse a **TODOSPECTRUM**. Envie **HOY MISMO** la tarjeta adjunta a la revista, que no necesita sobre ni franqueo. Deposítela en el buzón más cercano. Inmediatamente recibirá su primer ejemplar de **TODOSPECTRUM** más el **REGALO**.

Todospectrum
Bravo Murillo, 377
Tel. 733 79 69
28020 MADRID

```
        INC  HL
        CALL ^196E
        LD   A,(HL)
        PUSH AF
        LD   A,^80
        LD   (HL),A
        LD   HL,(PRILI)
        CALL ^196E
        LD   DE,(COMIE)
        PUSH DE
REBUC   LD   A,(HL)
        CP   ^80
        JR   Z,REEND
        POP  DE
        LD   (HL),D
        INC  HL
        LD   (HL),E
        DEC  HL
        PUSH HL
        LD   HL,(INCR)
        ADD  HL,DE
        EX   (SP),HL
        CALL ^19B8
        EX   DE,HL
        JR   REBUC
REEND   POP  DE
        POP  AF
        LD   (HL),A
        RET
SCAN    LD   HL,(^5C53)
OTRALI  LD   A,(HL)
        AND  ^C0
        RET  NZ
        INC  HL
        INC  HL
        LD   C,(HL)
        INC  HL
        LD   B,(HL)
        PUSH HL
        ADD  HL,BC
        LD   (^5C4D),HL
        POP  HL
        LD   (DCOMI),HL
        LD   (^5C5D),HL
        JR   NXST
OTRST   POP  HL
        LD   D,^02
        CALL EASTMT
        JR   NC,NXST
        LD   HL,(^5C4D)
        INC  HL
        JR   OTRALI
NXST    CALL NEXTCH
        PUSH HL
        LD   HL,RETAB
        LD   C,A
        CALL ^16DC
        JR   NC,OTRST
        LD   D,^00
        LD   E,(HL)
        ADD  HL,DE
        CALL ^162C
        JR   OTRST
RETAB   DEFB ^F8,RSAVE-$
        DEFB ^1C,RON-$
        DEFB ^EC,RRES-$
        DEFB ^ED,RRES-$
        DEFB ^E1,RRES-$
        DEFB ^F0,RRES-$
        DEFB ^F7,RRES-$
        DEFB ^E5,RRES-$
        DEFB ^00
RSAVE   LD   BC,^FFCA
SAON    LD   HL,(^5C5D)
SAON1   INC  HL
        LD   A,(HL)
        CALL ^18B6
        CP   ^0D
        RET  Z
        AND  B
        CP   C
        JR   NZ,SAON1
        LD   (^5C5D),HL
        JR   RRES
RON     LD   BC,^FEEC
        CALL SAON
OTRPA   RST  ^20
        CALL ^1C82
        PUSH AF
        RST  ^28
        DEFB ^02,^38
        POP  AF
        CP   ^2C
        RET  NZ
        CALL RRES
        JR   OTRPA
RRES    LD   HL,(^5C5D)
RES1    INC  HL
        LD   A,(HL)
        CP   ^0E
```



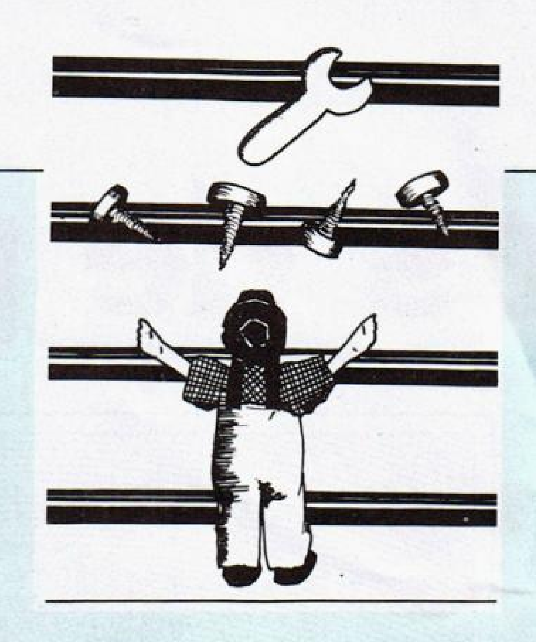
```
1625          JR    Z,RES2
1626          CALL  ^2D1B
1627          RET   C
1628          JR    RES1
1629 RES2     PUSH  HL
1630          CALL  ^18B6
1631          CP    ^3A
1632          JR    Z,RES3
1633          CP    ^0D
1634          JR    Z,RES3
1635          CP    ^2C
1636          POP   DE
1637          RET   NZ
1638          PUSH  DE
1639 RES3     DEC   HL
1640          DEC   HL
1641          LD    B,(HL)
1642          DEC   HL
1643          LD    C,(HL)
1644          CALL  NNL
1645          POP   HL
1646          RET   C
1647          PUSH  IX
1648          POP   BC
1649          PUSH  HL
1650          INC   HL
1651          INC   HL
1652          INC   HL
1653          LD    (HL),C
1654          INC   HL
1655          LD    (HL),B
1656          POP   HL
1657          LD    DE,(^5C5D)
1658          INC   DE
1659          LD    (^5C5B),HL
1660          PUSH  BC
1661          CALL  ^19E5
1662          LD    A,^FF
1663          CALL  ^1601
1664          POP   BC
1665          CALL  ^1A1B
1666          LD    HL,(^5C4D)
1667          LD    DE,(DCOMI)
1668          AND   A
1669          SBC   HL,DE
1670          EX    DE,HL
1671          LD    (HL),D
1672          DEC   HL
1673          LD    (HL),E
1674          RET
1675 NNL      LD    HL,(ULTLI)
1676          AND   A
1677          SBC   HL,BC
1678          RET   C
1679          LD    HL,(PRILI)
1680          SBC   HL,BC
1681          RET   Z
1682          CCF
1683          RET   C
1684          ADD   HL,BC
1685          LD    IX,(COMIE)
1686          PUSH  BC
1687          CALL  ^196E
1688 OTHLI    POP   BC
1689          CALL  ^1980
1690          RET   NC
1691          LD    DE,(INCR)
1692          ADD   IX,DE
1693          PUSH  BC
1694          CALL  ^19B8
1695          EX    DE,HL
1696          JR    OTHLI
1697 DCOMI    DEFW  ^0000
1698 ULTLI    DEFW  10000
1699 PRILI    DEFW  1
1700 COMIE    DEFW  10
1701 INCR     DEFW  10
1702 NEW      LD    HL,(^5CB2)
1703          LD    (HL),^3E
1704          DEC   HL
```

```
         LD    SP,HL
         DEC   HL
         DEC   HL
         LD    (^5C3D),HL
         LD    IY,^5C3A
         LD    HL,^5CB6
         LD    (^5C4F),HL
         LD    DE,^15AF
         LD    BC,^0015
         EX    DE,HL
         LDIR
         EX    DE,HL
         DEC   HL
         LD    (^5C57),HL
         INC   HL
         LD    (^5C53),HL
         LD    (^5C4B),HL
         LD    (HL),^80
         INC   HL
         LD    (^5C59),HL
         LD    (HL),^0D
         INC   HL
         LD    (HL),^80
         INC   HL
         LD    (^5C61),HL
         LD    (^5C63),HL
         LD    (^5C65),HL
         LD    A,^0F
         LD    (^5C8D),A
         LD    (^5C8F),A
         LD    (^5C48),A
         LD    A,^01
         OUT   (^FE),A
         DEC   (IY-58)
         DEC   (IY-54)
         LD    HL,PROUT
         LD    (^5CC5),HL
         LD    HL,^15C6
         LD    DE,^5C10
         LD    BC,^000E
         LDIR
         SET   1,(IY+1)
         CALL  ^0EDF
         LD    (IY+49),^02
         CALL  CLS
         XOR   A
         LD    DE,MCOPYR
         CALL  ^0C0A
         LD    (IY+1),^00
         SET   5,(IY+2)
         JP    MAIN1
MCOPYR   DEFB  ^80
         DEFM  "c 1984 Arana SOFTWAR"
         DEFB  ^C5
COMND    RST   8
         DEFB  ^0B
PROUT    CALL  PCALC
         PUSH  HL
         PUSH  AF
         LD    A,(HL)
         LD    (PAMOD+1),A
         POP   AF
         CP    ^20
         JR    NC,PROU2
         CP    ^10
         JR    C,PROU2
         CP    ^18
         JR    C,PROU1
         SUB   ^18
         LD    DE,TOTAB
         CALL  POTOK
         POP   HL
         RET
PROU1    POP   HL
         JR    PROUT1
PCALC    LD    HL,PRDAT
         BIT   1,(IY+1)
         RET   NZ
         INC   HL
         BIT   0,(IY+2)
         RET   Z
         INC   HL
         RET
PROU2    LD    HL,(PRDIR)
         CALL  ^162C
         POP   HL
PSTORE   LD    A,(PAMOD+1)
         LD    (HL),A
         RET
PRDIR    DEFW  PROUT1
PRDAT    DEFB  ^00,^00,^00
POTOK    PUSH  AF
         CALL  ^0C41
         LD    A,^20
         BIT   0,(IY+1)
         CALL  Z,^0C3B
POEACH   LD    A,(DE)
         AND   ^7F
         CALL  ^0C3B
         LD    A,(DE)
         INC   DE
         ADD   A,A
         JR    NC,POEACH
         POP   DE
         CP    ^48
         JR    Z,PROU3
```



```
        CP    ^82
PROU3   LD    A,^20
        JP    ^QC3B
PROUT1  CALL  ^0B03
        CP    ^20
        JP    NC,POABLE
        CP    ^10
        JR    C,PCTRL
        CP    ^18
        JR    NC,PINTER
        CP    ^16
        JR    NC,P2OPER
        LD    DE,PCONT
POTV1   JP    ^0A7D
PCONT   LD    DE,PROUT
        CALL  ^0A80
        LD    HL,(^5C0E)
        LD    D,A
        LD    A,(PRDIR+1)
        CP    ^09
        JP    Z,^0A91
        LD    A,L
        CP    ^16
        JP    C,^2211
        JR    NZ,POTAB
        LD    B,H
        LD    C,D
        LD    A,^3F
        SUB   C
        JP    C,^0AAC
        ADD   A,^04
        LD    C,A
        XOR   A
        SRL   C
        JR    C,PATINP
        LD    A,^15
PATINP  LD    (PAMOD+1),A
        CALL  ^0AA3
        CALL  PCALC
        JP    PSTORE
POTAB   LD    A,H
POFILL  CALL  ^0B03
        PUSH  AF
        LD    A,(PAMOD+1)
        CP    ^01
        RL    C
        POP   AF
        ADD   A,C
        SUB   ^03
        AND   ^3F
        JP    ^0ACA
P2OPER  LD    DE,POTV2
        JR    POTV1
POTV2   LD    DE,PCONT
        JP    ^0A70
PINTER  LD    A,^3F
        JR    POABLE
PCTRL   CP    ^06
        JR    NZ,POBACK
        SLA   C
        LD    A,^44
        SUB   C
        AND   ^30
        ADD   A,^10
        JR    POFILL
POBACK  CP    ^08
        JR    NZ,PRIGHT
        LD    A,(PAMOD+1)
        AND   A
        JR    Z,PBCK1
        XOR   A
        LD    (PAMOD+1),A
        RET
PBCK1   LD    A,^15
        LD    (PAMOD+1),A
        INC   C
        LD    A,^22
        CP    C
        JR    NZ,PBCK3
        BIT   1,(IY+1)
```

```
        JR    NZ,PCBK2
        INC   B
        LD    C,^02
        LD    A,^19
        CP    B
        JR    NZ,PBCK3
        DEC   B
PCBK2   LD    C,^21
        XOR   A
        LD    (PAMOD+1),A
PBCK3   JP    ^0DD9
PRIGHT  CP    ^09
        JR    NZ,PENTER
        LD    A,(IY+87)
        PUSH  AF
        LD    (IY+87),A
        LD    A,^20
        CALL  POCHAR
        POP   AF
        LD    (IY+87),A
        JP    ^0ADC
PENTER  CP    ^0D
        JR    NZ,PINTER
        BIT   1,(IY+1)
        JP    NZ,CBUFF
        XOR   A
        LD    (PAMOD+1),A
        LD    C,^21
        CALL  ^0C55
        DEC   B
        JP    ^0DD9
POABLE  CALL  POANY
        JP    ^0ADC
POANY   CP    ^90
        JR    C,POCHAR
        SUB   ^A5
        JP    NC,^0B5F
        ADD   A,^15
        PUSH  AF
        LD    A,(PAMOD+1)
        AND   A
        JR    Z,PUDGSP
        LD    A,^20
        CALL  POCHAR
PUDGSP  POP   AF
        PUSH  BC
        LD    BC,(^5C7B)
        JP    ^0B6A
POCHAR  PUSH  BC
        LD    BC,CHTAB-^100
        EX    DE,HL
        LD    HL,^5C3B
        RES   0,(HL)
        CP    ^20
        JR    NZ,POCH3
        SET   0,(HL)
POCH3   LD    H,^00
        LD    L,A
        ADD   HL,HL
        ADD   HL,HL
        ADD   HL,HL
        ADD   HL,BC
        POP   BC
        EX    DE,HL
        LD    A,(PAMOD+1)
        AND   A
        JR    Z,PRALL
        XOR   A
        LD    (PAMOD+1),A
        JR    PALL7
PRALL   LD    A,C
        DEC   A
        LD    A,^21
        JR    NZ,PALL1
        DEC   B
        LD    C,A
        BIT   1,(IY+1)
        JR    Z,PALL1
        PUSH  DE
        CALL  CBUFF
        POP   DE
        LD    A,C
PALL1   CP    C
        PUSH  DE
        CALL  Z,^0C55
        POP   DE
        LD    A,^15
        LD    (PAMOD+1),A
PALL7   PUSH  BC
        PUSH  HL
        LD    A,(IY+87)
        LD    B,^FF
        RRA
        JR    C,PALL2
        INC   B
PALL2   RRA
        RRA
        SBC   A,A
        LD    C,A
        LD    A,^08
        AND   A
        BIT   1,(IY+1)
        JR    Z,PALL3
        SET   1,(IY+48)
        SCF
PALL3   EX    DE,HL
```



```
PALL4   EX    AF,AF'
PAMOD   JR    PSM
        LD    A,^0F
        AND   C
        LD    C,A
        LD    A,^F0
        OR    B
        LD    B,A
        XOR   A
        RRD
        LD    A,(DE)
        AND   B
        XOR   (HL)
        XOR   C
        LD    (DE),A
        XOR   A
        RLD
        JR    PTEXP
PSM     LD    A,^F0
        AND   C
        LD    C,A
        LD    A,^0F
        OR    B
        LD    B,A
        LD    A,(DE)
        AND   B
        XOR   (HL)
        XOR   C
        LD    (DE),A
PTEXP   EX    AF,AF'
        JR    C,PALL6
        INC   D
PALL5   INC   HL
        DEC   A
        JR    NZ,PALL4
        EX    DE,HL
        DEC   H
        BIT   1,(IY+1)
        CALL  Z,^0BDB
        POP   HL
        POP   BC
        LD    A,(PAMOD+1)
        AND   A
        RET   NZ
        DEC   C
        INC   HL
        RET
PALL6   EX    AF,AF'
        LD    A,^20
        ADD   A,E
        LD    E,A
        EX    AF,AF'
        JR    PALL5
TOTAB   DEFB  ^BF
        DEFM  "ELS"
        DEFB  ^C5
        DEFM  "ON ER"
        DEFB  ^D2
        DEFM  "AUT"
        DEFB  ^CF
        DEFM  "RENU"
        DEFB  ^CD,"O",^CE
        DEFM  "SWA"
        DEFB  ^D0
        DEFM  "MOD"
        DEFB  ^C5,^C0
CBUFF   LD    HL,PAMOD+1
        LD    (HL),^00
        JP    ^0ECD
```

Trucos de progra



Algunas personas afirman que la programación de ordenadores es un arte más que una ciencia y que sólo unos «elegidos» son capaces de programar sacando el máximo partido a la máquina. Sin llegar a eso, ni mucho menos, sí que es cierto que existen muchos «trucos» a la hora de programar que no se explican en el manual y que sin embargo pueden simplificar mucho los programas al ahorrarnos el escribir muchas líneas.

Con este artículo pretendemos explicar los más comunes, de modo que sus programas puedan usarlos y por tanto ser más cortos y lo que es más importante, más rápidos.

Las funciones lógicas

Normalmente se piensa que las funciones lógicas AND y OR sólo tienen sentido dentro de un IF para averiguar si una condición es cierta o falsa. Lo mismo sucede con los símbolos de comparación (=, ^{4}, ^{5}, ^{45}, etc.) que normalmente sólo son usados dentro de instrucciones de este tipo. Pero su uso es mucho más amplio y nos pueden ahorrar muchos IF como veremos a continuación.

Cuando en un IF se escribe una expresión, lo que se le está diciendo al ordenador es que la evalúe y dé un resultado, y entonces el IF ejecuta lo que viene a continuación del THEN si el resultado es distinto de cero, y no lo hace si el resultado es cero. Por tanto la expresión IF 0 THEN..., no ejecutaría nunca la parte condicionada (la que está detrás del THEN) al ser el resultado del IF siempre cero, la situación opuesta de la anterior sería IF 1 THEN..., que se ejecutaría siempre debido a que el resultado del IF siempre es distinto de cero. Esto nos lleva al primer «truco», que consiste en un sistema de ahorrar tecleo y argumentar la velocidad de un programa cuando se está realizando una operación con todos los elementos de una matriz que no valgan cero. Normalmente se haría así:

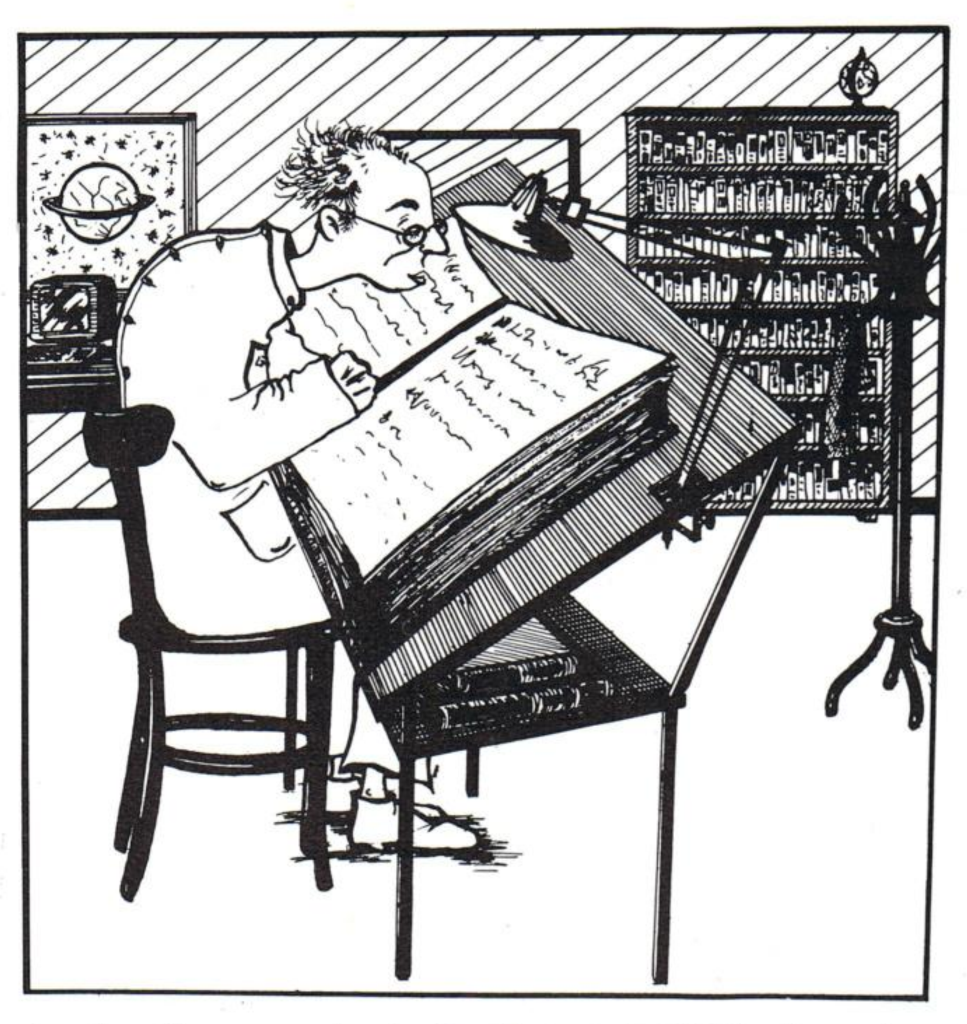
10 FOR I=1 TO 100 (si la matriz es de cien elementos)
20 IF A(I)< >0 THEN ...operaciones a realizar...
30 NEXT I

Pero se puede simplificar así:

10 FOR I=1 TO 100
20 IF A(I) THEN ...operaciones a realizar...
30 NEXT I

La razón de que quitemos la comparación se debe a que ésta devuelve un resultado no cero cuando se cumple (es decir, cuando el elemento de la matriz no vale cero) y cero cuando no se cumple (cuando el elemento de la matriz vale cero), pero esto es lo mismo que tiene almacenado la matriz y por tanto nos podemos ahorrar la comparación, ya que el valor no importa para ver si ejecuta la parte condicionada, porque el or-

…mación en BASIC



denador sólo comprueba si vale cero.

El caso opuesto a esta comparación sería:

```
10 FOR I=1 TO 100
20 IF A(I)=0 THEN ...operaciones a realizar...
30 NEXT I
```

En este caso sólo se ejecuta la parte condicionada cuando el elemento de la matriz valga cero. Usted ya estará pensando ¡aja, aquí no se puede abreviar ya que queremos hacer justo lo contrario, es decir, ejecutar cuando valga cero! Pero por suerte el Spectrum dispone de otra función para estos casos. Se trata de NOT, y lo que hace es convertir cualquier número que no valga cero en cero y si el número vale cero lo convierte en uno. Para decirlo de otro modo, invierte el valor. Por ello podemos escribir:

```
10 FOR I=1 TO 100
20 IF NOT A(I) THEN ...operaciones a realizar...
30 NEXT I
```

Y en este caso las operaciones se ejecutarán cuando el elemento valga cero, ya que al ponerle el NOT delante, el resultado final será uno en ese caso y cero en los demás.

Pero en muchas ocasiones nos podemos ahorrar hasta el IF. Este es el caso de ciertas situaciones en las que una variable tiene que adoptar un valor si otra es cero y otro totalmente distinto si la segunda es distinta de cero. Imaginemos que necesitamos que «A» valga 3 si «B» es cero y si no que adopte el valor 2*B. En un principio parece que se podría poner A=2*B. Pero cuando B valga 0, A valdrá 0 y no 3 como queremos. Parecería que aquí hay que poner un IF para hacer esa asignación sólo si B es distinto de cero y hacer A=3 en caso contrario. Pero no tiene por qué ser así.

Las comparaciones entre números (=, <, >, <>, etc.) dan como resultado 1 si la relación indicada se cumple y 0 si no es así. Esto nos permite escribir la operación anterior como A=2*B*(B<>0) + 3*(B=0). Ya que cuando B valga cero, B <> 0 será falso, B=0 cierto y, por tanto, el valor de A será A=2*B*0+3*1, o lo que es lo mismo, A=3. En cambio cuando B sea distinto de cero, obtendremos A=2*B*1+3*0, es decir, A=2*B. La expresión podría haberse simplificado poniendo A=2*B+3*(B=0), ya que la comparación B<>0 es redundante como puede observarse volviendo a evaluar esta expresión en ambos casos.

Otro caso que se puede simplificar es el de seleccionar el mayor de dos valores. Esto normalmente se hace con IF, dentro del cual se pone la comparación a realizar. Pero otro método es hacer:

```
MAX=A*(A>B)+B*(A<=B)
```

La razón de que en la segun-

da comparación se ponga menor o igual en lugar de menor solamente se debe a que si ambos números son iguales y sólo hubiéramos puesto menor, ambas comparaciones serían falsas y por tanto MAX adoptaría el valor cero. En cambio tal como lo hemos puesto, cuando A y B sean iguales MAX adoptará el valor de B. El símbolo igual también podía haberse puesto en la primera comparación, pero nunca en las dos a la vez.

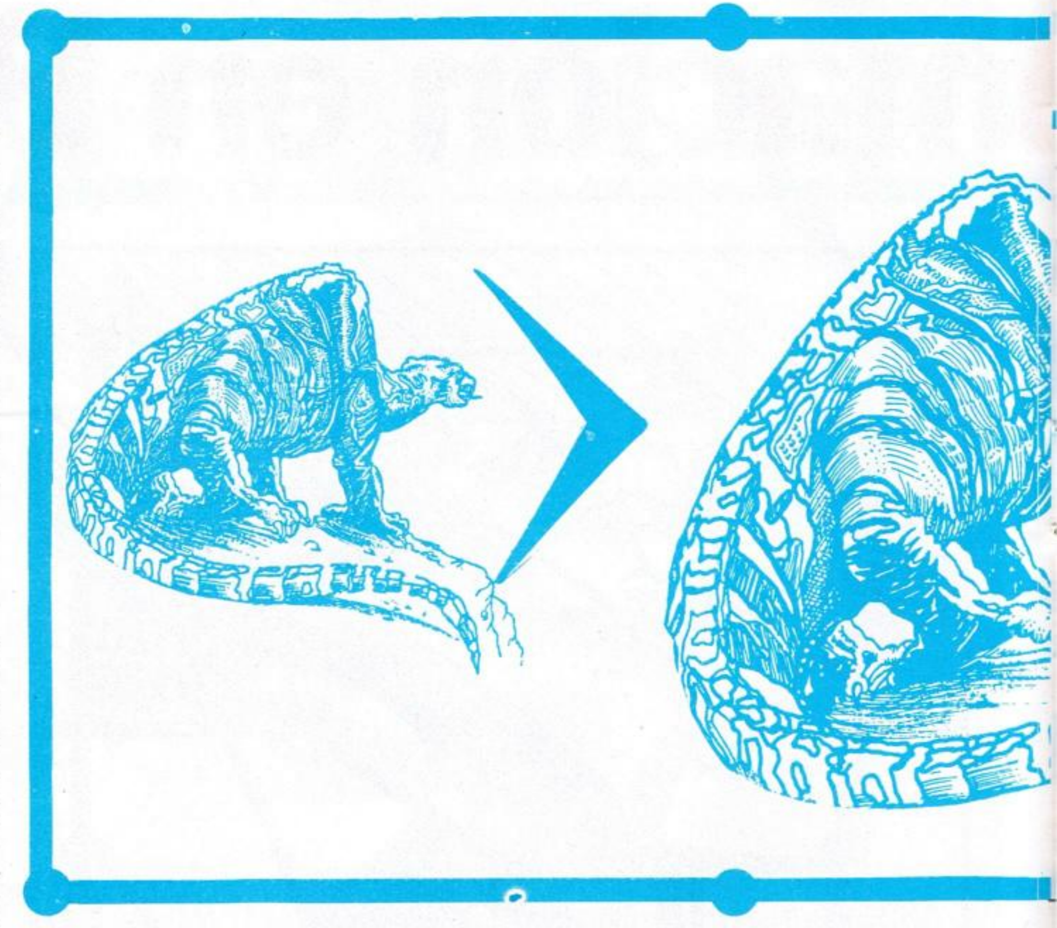
Operadores lógicos

Usted seguramente habrá usado las operaciones +, –, * y / en sus programas. Estos se denominan operadores matemáticos y como su propio nombre indica, realizan operaciones matemáticas con números. Según esta definición, los operadores lógicos serán aquellos que realicen operaciones lógicas con números. Aunque esto puede sonar extraño, en realidad es muy sencillo, ya que una operación lógica es aquella que sólo distingue entre cero y cualquier otro valor distinto de cero.

Ya hemos visto un operador lógico, el NOT, y su modo de funcionamiento. Su característica más importante es que opera sobre un solo número invirtiéndolo, de modo que si es cero lo convierte en uno y si es distinto de cero lo convierte a cero.

Además de esta operación existen otras dos que operan con dos números. La AND y la OR. Ambas siguen el mismo formato de las operaciones matemáticas, es decir, el primer operando, la operación y el segundo operador.

La operación AND (Y en inglés) devuelve un valor distinto de cero si el primer operando es distinto de cero Y el segundo



A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

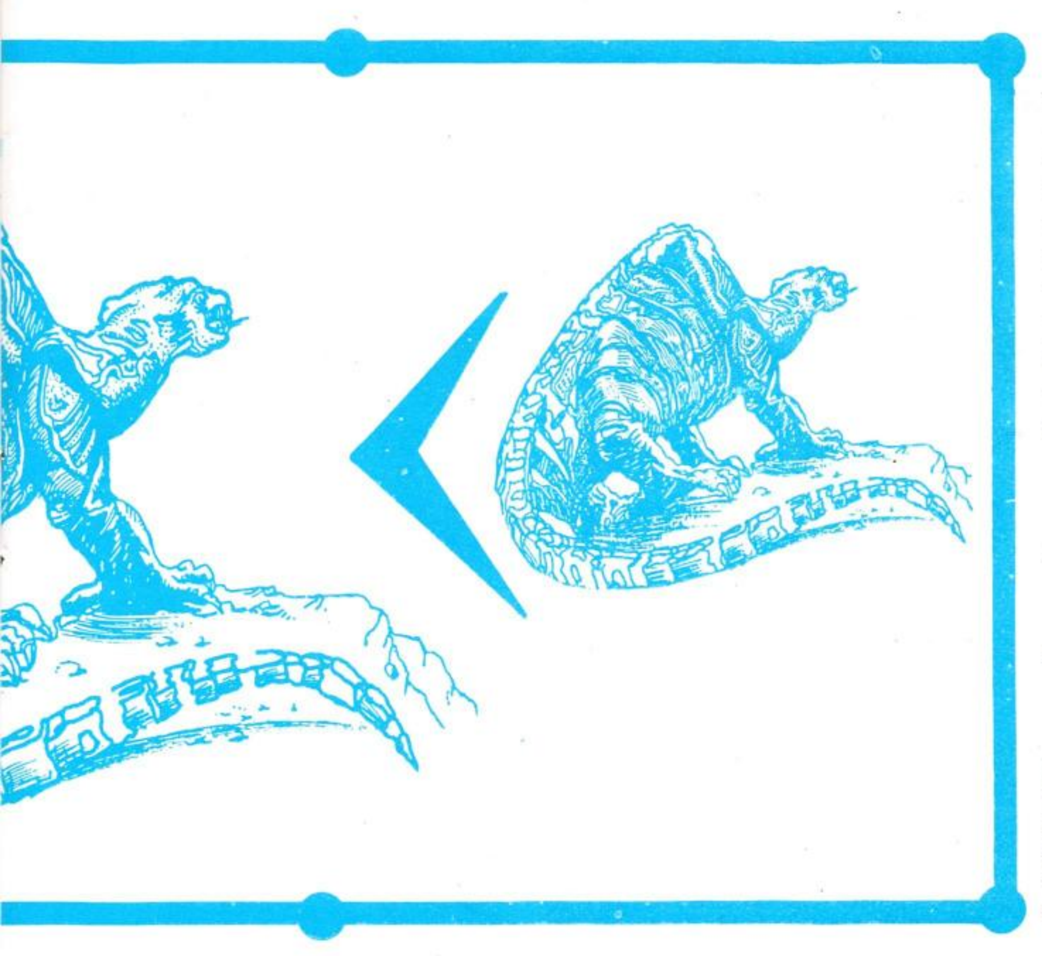
Figura 1

A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

Figura 2

también. O dicho de otro modo, cuando alguno de los dos operandos es cero (o los dos) entonces el resultado es cero. Esto se puede representar con una tabla de verdad como se muestra en la figura 1, en la que se representa el valor de cada uno de los operadores y el valor del AND de ambos. Hay que tener en cuenta que en esta tabla cuando ponga "1" quiere decir «cualquier número distinto de cero». Esto resulta especialmente importante en el valor devuelto, ya que como vimos antes, las comparaciones devolvían un valor 1 ó 0, pero en este caso no necesariamente devuelve un 1 si ambos son distintos de cero (para ser exactos la mayoría de las veces no devuelve 1), sino que puede devolver cualquier número distinto de cero.

Si necesitamos asegurarnos de que este valor sea uno o cero, se puede usar la siguiente expresión:



ABS(SGN(A AND B))

Donde A y B pueden ser números u otras expresiones. Curiosamente se puede escribir esta expresión con igualdades:

(A< >0)*(B< >0)

La operación OR tiene un funcionamiento similar a la anterior pero devuelve un número distinto de cero cuando cualquiera de los dos operadores, o los dos, son distintos de cero. La tabla de verdad se da en la figura 2. En ella podemos ver que sólo se devuelve un cero cuando los dos operandos son cero. Al igual que con el NAD, un uno significa «cualquier valor distinto de cero», por lo que para obtener un 1 tendremos que hacer

ABS(SGN(A OR B))

Donde A y B, como antes, son números o expresiones. Esta expresión también se puede reescribir con comparaciones tal como sigue:

SGN((A< >0)+(B< >0))

El FOR condicional

Una de las primeras expresiones que se aprende en el BASIC son los bucles FOR-...NEXT. Su indudable utilidad y su facilidad de uso lo convierten en una de las expresiones más usadas en cualquier tipo de programas. Pero tiene algunos problemas que la hacen inusable en determinados casos, como cuando se quieren examinar todos los elementos de una matriz y si alguno es cero, entonces se termina sin comprobar los restantes.

Para solucionar esto y poder seguir usando el FOR, existen dos soluciones. Ninguna de ellas es «elegante» y a algún programador profesional no le puede agradar el método, pero es indudable que funcionan.

La primera solución para salir antes de tiempo del bucle es hacer un GO TO a la instrucción siguiente al NEXT. Supongamos que queremos examinar una matriz A de 100 elementos y salir cuando encontremos alguno que valga cero, pero existe la posibilidad de que no haya ninguno que valga cero. Esto lo podemos hacer como sigue:

```
10 FOR I=1 TO 100
20 IF A(I)=0 THEN GO TO 50
30 NEXT I
40 si llega aquí no hay ninguno que valga cero.
50 resto de las instrucciones.
```

Este método tiene la ventaja de que en I se conserva el número del elemento que es cero. Por otra parte el GO TO, sobre todo en programas largos, es algo lento.

El segundo sistema es más rápido al prescindir del GO TO y además sale correctamente del bucle, aunque no conserva el número del elemento que es cero.

```
10 FOR I=1 TO 100
20 IF A(I)=0 THEN I=100
30 NEXT I
40 resto de las instrucciones.
```

Al ser la comparación cierta se hace que I valga cien y al llegar al NEXT el ordenador se cree que ya ha hecho el bucle cien veces y sale. Como hemos dicho en este caso no se conserva el valor de I, a menos que usemos otra variable auxiliar para almacenarlo, pero la salida se hace correctamente y de un modo más rápido. Además si hay más instrucciones entre el IF y el NEXT las ejecuta, mientras que con el modo anterior no lo hacía.

Esperamos que con estos pequeños «trucos» los programas os salgan más pequeños y rápidos.



Suplemento especial Enero 1986

Radiografía del QL

Gráficos con SuperBasic: Colores

DIBUJANDO CON SUPERBASIC

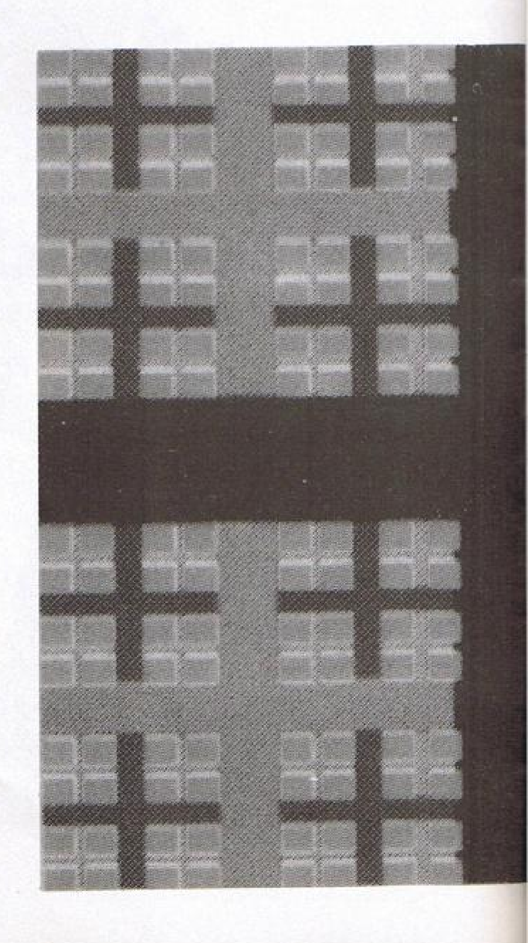
En la primera parte de este artículo vimos cómo realizar diversos dibujos utilizando los comandos disponibles en el BASIC de nuestro QL, pero para que realmente sean vistosos hace falta añadirles un toque de color.

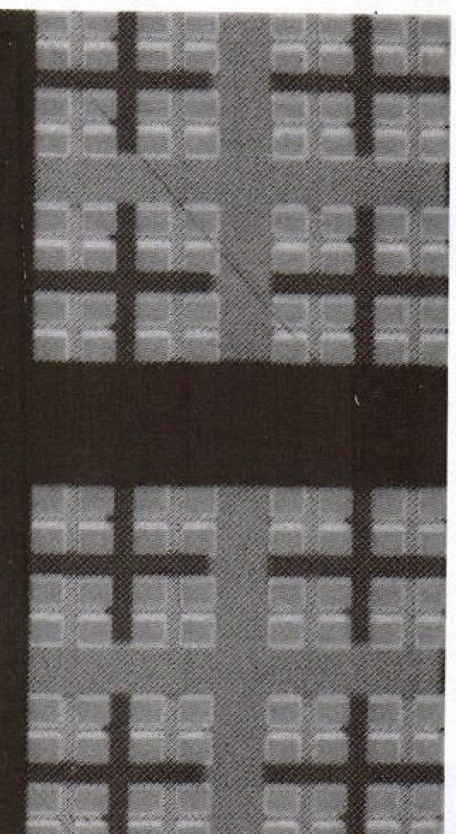
La instrucción más conocida que realiza esto es INK n, que nos permite fijar el color con que se hacen todos los dibujos a partir de que se ejecute esta instrucción. El rango de n va del 0 al 7 (con MODE 8 activo) y cada uno de los valores indica un color distinto. Como ejemplo de su uso se da el listado de la figura 1, que va dibujando círculos desplazados hacia la derecha y cada uno de un color.

No obstante éste no es el comando más espectacular de todos los disponibles. Existe uno mucho más colorista, FILL. Del mismo modo que INK, actúa sobre las instrucciones que se den después, pero no modifica el color fijado por INK, sino que rellena áreas cerradas con ese color.

Su funcionamiento es muy sencillo, en primer lugar se teclea FILL 1 para activarlo, a continuación se dibuja una figura cerrada (que esté totalmente rodeada por líneas) por medio de las instrucciones LINE, CIRCLE o ELLIPSE y al dibujarla el ordenador, automáticamente la rellenará con el mismo color de la tinta. Por último, para desactivar esta modalidad se teclea FILL 0.

En listado 2 se ve la diferencia entre FILL 0 (el modo en el que normalmente está el QL al encenderlo) y FILL 1. En la primera parte del programa (líneas 110-130) se crea un cuadrado sin rellenar en la parte izquierda de la pantalla. En las líneas siguientes se vuelve a hacer lo mismo a la derecha de la pantalla pero en esta ocasión





se ha indicado FILL 1 y el cuadrado que se crea se rellena del mismo color.

Un ejemplo más elaborado del uso de este comando es el del listado 3, donde con unas sencillas instrucciones de trazado de arcos y el comando FILL, se logra dibujar unos labios sonriendo.

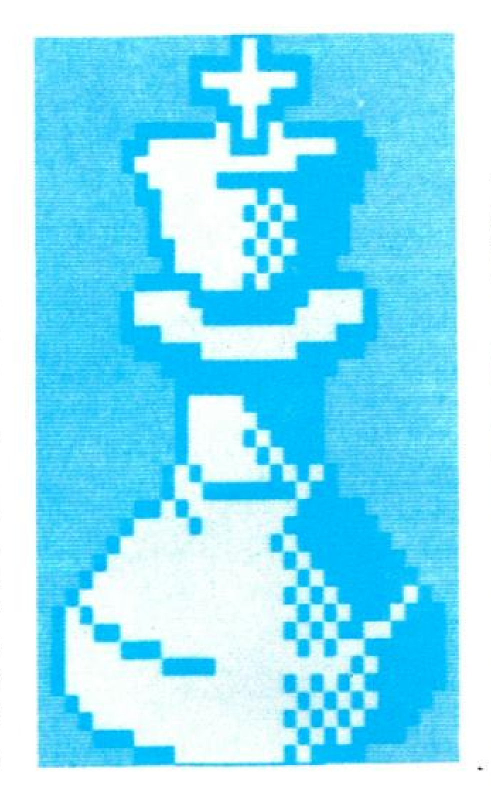
Cambiando el color

Hasta ahora todas las modificaciones de color se han hecho previamente a los dibujos, pero es posible hacerlas con posterioridad. Esto permite ver diversas combinaciones y hasta dotar de cierta animación a los dibujos. El comando utilizado para ello es RECOL, c0, c1, c2, c3, c4, c5, c6, c7. Donde c0, c1... c7 son los códigos de los distintos colores que se reasignan de nuevo, correspondiendo la combinación RECOL 0, 1, 2, 3, 4, 5, 6, 7 a la disposición al encender.

Por tanto si queremos que el que es blanco al encender (el séptimo) cambie súbitamente a rojo (el segundo) basta hacer: RECOL 0, 1, 2, 3, 4, 5, 6, 2 dejando los demás colores con su mismo número de orden. Aplicando esto al ejemplo de los labios, podemos dar la impresión de que éstos se abren y se cierran rápidamente por medio de un par de sencillas modificaciones al programa original de modo que quede como se indica en el listado 4.

Más cuadrados

Ya hemos visto antes un método de hacer rectángulos de colores, pero existe uno más fácil aún con el que no hace falta activar el FILL. Esto se consigue con el comando BLOCK an,al,y,x,c. Siendo "an" la anchura del rectángulo, "al" la altura, "x,y" la coordenada de la parte superior izquierda y "c" el color con que se quiere rellenar. En el listado 5 se encuentra un programa que hemos denominado ciudad,



ya que crea bloques rectangulares de diversa forma y tamaño de modo que parece una ciudad de rascacielos.

Algo más elaborado es el sexto listado, en el que hemos empleado una técnica denominada "recursión" y que no poseen la mayoría de los otros lenguajes. Esto nos permite diseñar procedimientos que se llaman a sí mismos. En este caso "caja" dibuja una cruz de un color y en cada uno de los cuatro cuadrados resultantes vuelve a llamarse a sí misma para crear otras cuatro cruces, que a su vez vuelven a llamar a caja.

```
100 CLS
110 FOR i=1 TO 110
120 INK i MOD 8
130 CIRCLE i,50,50
140 NEXT i
150 STOP
```

```
100 CLS
110 INK 6
120 FILL 1
130 LINE 5,5 TO 5,60 TO 60,60 TO 60,5 TO 5,5
140 FILL 0
150 LINE 65,5 TO 65,60 TO 120,60 TO 120,5 TO 65,5
160 STOP
```

```
100 PAPER 1:INK 2:CLS
110 FILL 1
120 ARC 30,50 TO 80,70,-1.5 TO 80,28,-.3 TO 30,50,-.75
130 FILL 0
131 FILL 1
132 ARC 130,50 TO 80,70,1.5 TO 80,28,.3 TO 130,50,.75
133 FILL 0
140 FILL 1:INK 7
150 ARC 35,50 TO 125,50,1.2 TO 35,50,1.2
160 FILL 0:INK 2
170 LINE 35,50 TO 125,50
180 FOR k=50 TO 110 STEP 10
190 LINE k,36 TO k,64
200 END FOR k
```

```
100 PAPER 1:INK 2:CLS
110 FILL 1
120 ARC 30,50 TO 80,70,-1.5 TO 80,28,-.3 TO 30,50,-.75
130 FILL 0
140 FILL 1
150 ARC 130,50 TO 80,70,1.5 TO 80,28,.3 TO 130,50,.75
160 FILL 0
170 FILL 1:INK 7
180 ARC 35,50 TO 125,50,1.2 TO 35,50,1.2
190 FILL 0:INK 2
200 LINE 35,50 TO 125,50
210 FOR k=50 TO 110 STEP 10
220 LINE k,36 TO k,64
230 END FOR k
240 RECOL 0,1,2,3,4,5,6,2
```

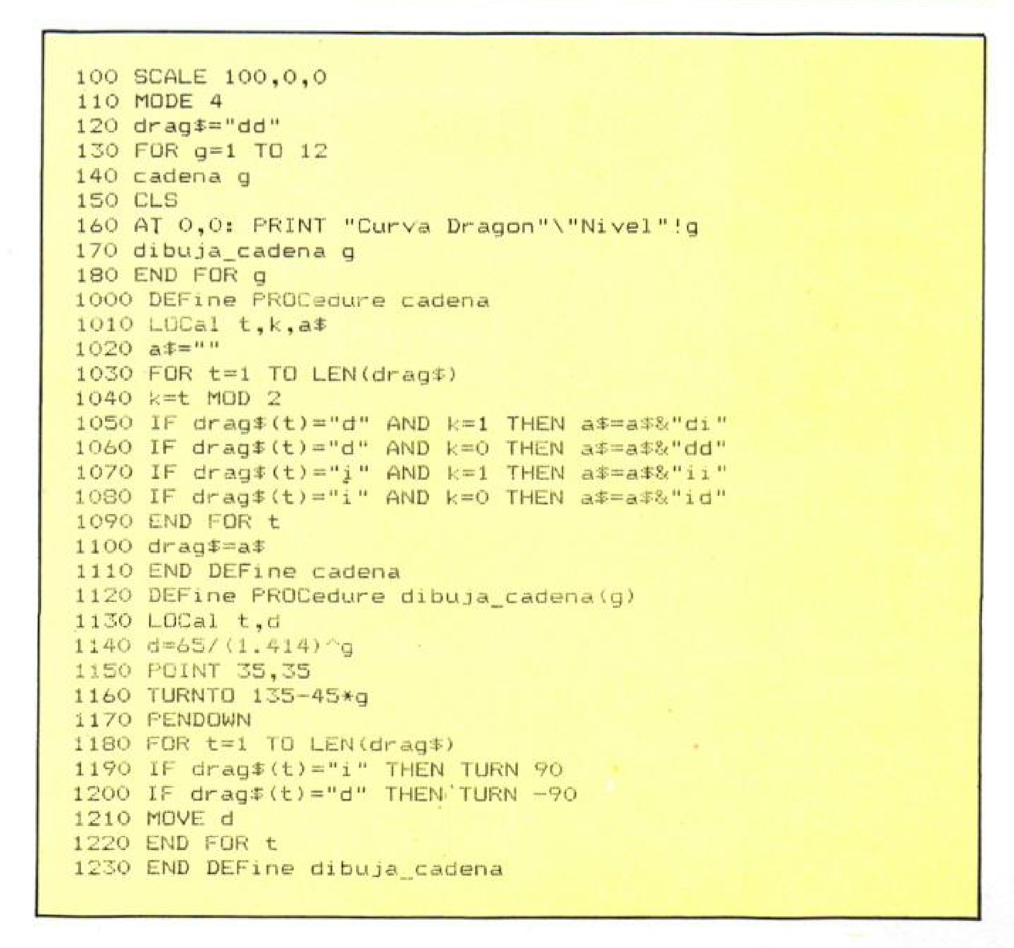
La tortuga

Otra de las habilidades de dibujo del QL es la denominada «tortuga». Idea que tienen otros lenguajes, como el Logo o el Smaltalk que consiste en que disponemos de una tortuga que tiene un lápjz y a la que podemos dar instrucciones de movimiento. Las principales son PENDOWN, que apoya el lápiz en el papel y, por tanto al moverse dibuja; PENUP que lo levanta; MOVE n, que avanza n puntos en la dirección en que esté mirando; TURN a, que gira a la derecha tantos grados como indique "a" desde la dirección actual y TURNTO a, que gira también "a" grados, pero no desde donde apunta "a", sino desde la posición en la que mira hacia arriba, que en este comando se considera la inicial. La diferencia entre TURN y TURNTO se puede ver en las figuras 7a y 7b. En ambas la tortuga está a 45 grados de la posición vertical (denominada origen) y tam-

```
100 color=RND(7)
110 altura= RND(10 TO 200)
120 anchura= RND(10 TO 160)
130 posicion=RND(280)
140 BLOCK anchura, altura,posicion,200-altura,color
150 GO TO 100
```

```
100 PAPER 7:CLS
110 caja 100,10,180,0
120 STOP
130 DEFine PROCedure caja (x,y,s,n)
140   LOCal a,b,x0,y0,s0
150   IF n=5 THEN RETurn
160   BLOCK 1.6*s,s,x,y,n
170   FOR a=0 TO 1
180     FOR b=0 TO 1
190        x0=x+a*6.4*s/7
200        y0=y+b*4*s/7
210        s0=3*s/7
220        caja x0,y0,s0,n+1
230     END FOR b
240   END FOR a
250 RETurn
260 END DEFine caja
```

```
100 POINT 10,10
110 PENDOWN
120 FOR i=1 TO 4
130   MOVE 50
140   TURN 90
150 END FOR i
```



```
100 SCALE 100,0,0
110 MODE 4
120 drag$="dd"
130 FOR g=1 TO 12
140 cadena g
150 CLS
160 AT 0,0: PRINT "Curva Dragon"\"Nivel"!g
170 dibuja_cadena g
180 END FOR g
1000 DEFine PROCedure cadena
1010 LOCal t,k,a$
1020 a$=""
1030 FOR t=1 TO LEN(drag$)
1040 k=t MOD 2
1050 IF drag$(t)="d" AND k=1 THEN a$=a$&"di"
1060 IF drag$(t)="d" AND k=0 THEN a$=a$&"dd"
1070 IF drag$(t)="i" AND k=1 THEN a$=a$&"ii"
1080 IF drag$(t)="i" AND k=0 THEN a$=a$&"id"
1090 END FOR t
1100 drag$=a$
1110 END DEFine cadena
1120 DEFine PROCedure dibuja_cadena(g)
1130 LOCal t,d
1140 d=65/(1.414)^g
1150 POINT 35,35
1160 TURNTO 135-45*g
1170 PENDOWN
1180 FOR t=1 TO LEN(drag$)
1190 IF drag$(t)="i" THEN TURN 90
1200 IF drag$(t)="d" THEN TURN -90
1210 MOVE d
1220 END FOR t
1230 END DEFine dibuja_cadena
```

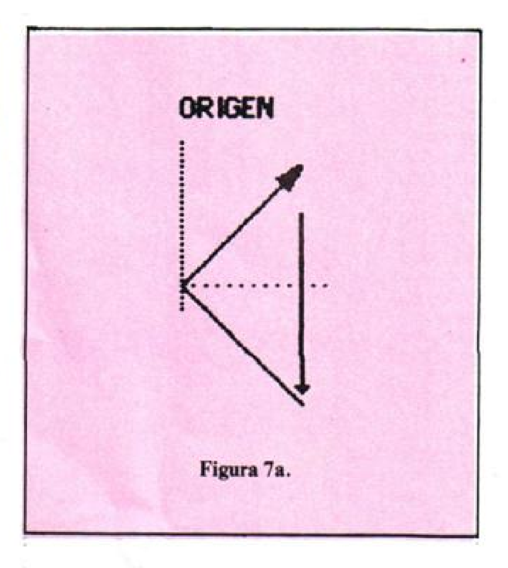


Figura 7a.

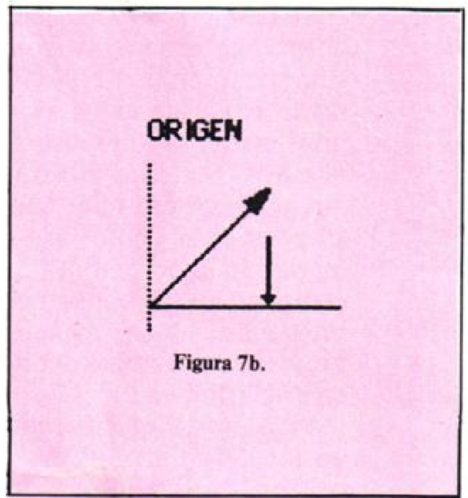
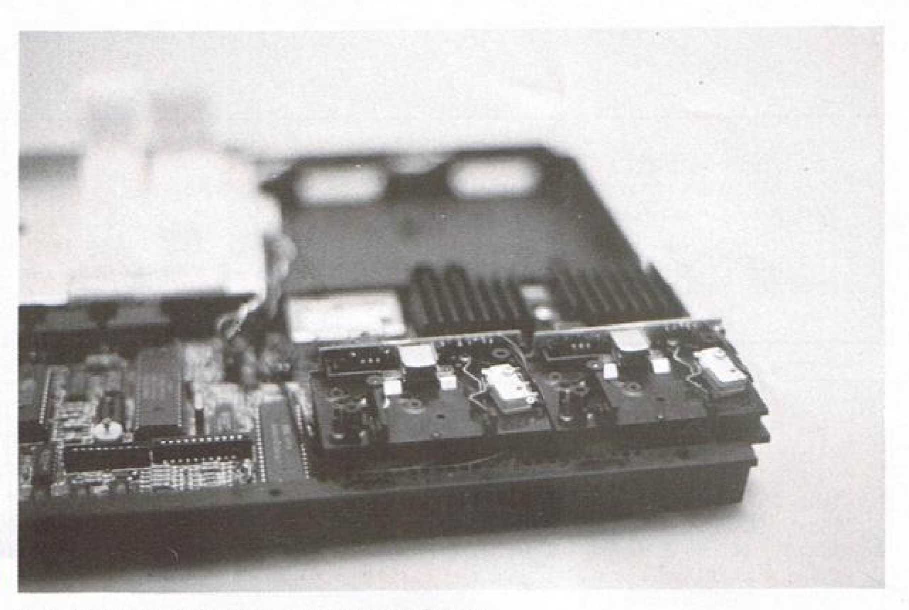


Figura 7b.

bién en ambas se le manda girar 90 grados. La diferencia estriba en que en el primer caso se le dice TURN, quedando a 90 grados de la posición anterior y a 135 del origen, mientras que en la segunda se dice TURNTO y queda a 90 grados del origen y a 45 de la posición anterior.

En la figura 8 se da un sencillo listado para hacer que nuestra tortuga dibuje un cuadrado y por último en la figura 9 se utiliza la tortuga de nuevo para crear la denominada curva del dragón, un efecto realmente impresionante conseguido con muy pocas líneas de programa.

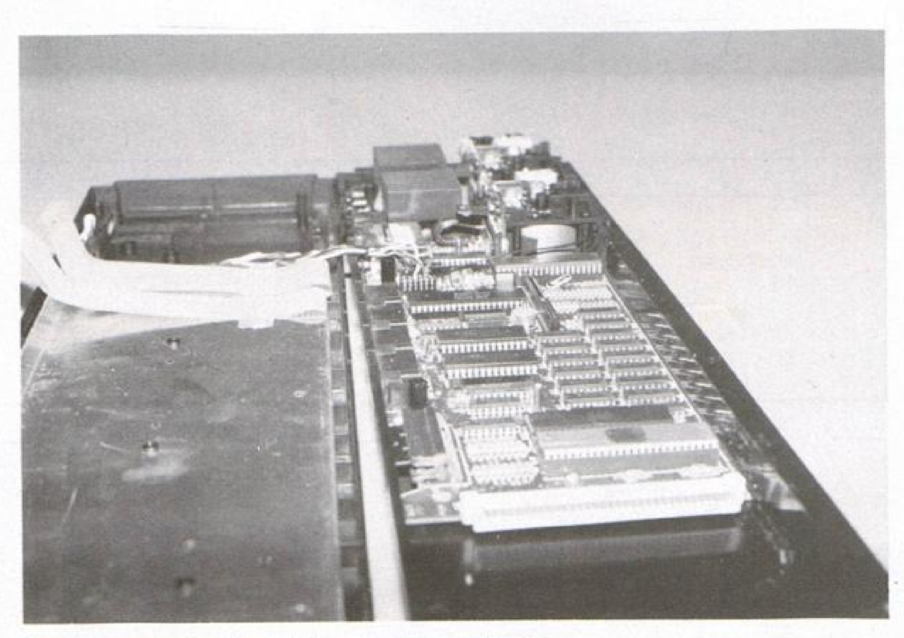
Radiografía al QL: Descripción de su Funcionamiento



Detalle de los *microdrives*. Estas unidades se encuentran situadas a la derecha de la placa principal. Detrás se puede ver el radiador de la fuente de alimentación.

Con un tamaño más de teclado que de ordenador completo y envuelto en una bonita carcasa de plástico negro, el QL es una máquina que asombra a muchos por las posibilidades que ofrece. Pero para hacer posible esto, la gente de Sinclair ha tenido que trabajar mucho y crear un nuevo concepto totalmente distinto a los realizados previamente, como explicamos en este artículo.

Para abrir el ordenador se quitan diez tornillos dejando el ordenador dividido en dos partes. Por un lado el teclado, que va cubierto de una placa metálica, y en la parte inferior los *microdrives,* la fuente de alimentación y la placa de circuito impreso que constituye el corazón del ordenador.



En esta perspectiva desde la izquierda se puede ver un primer plano del conector de expansión al que se conectan las unidades de disco y las ampliaciones de memoria.

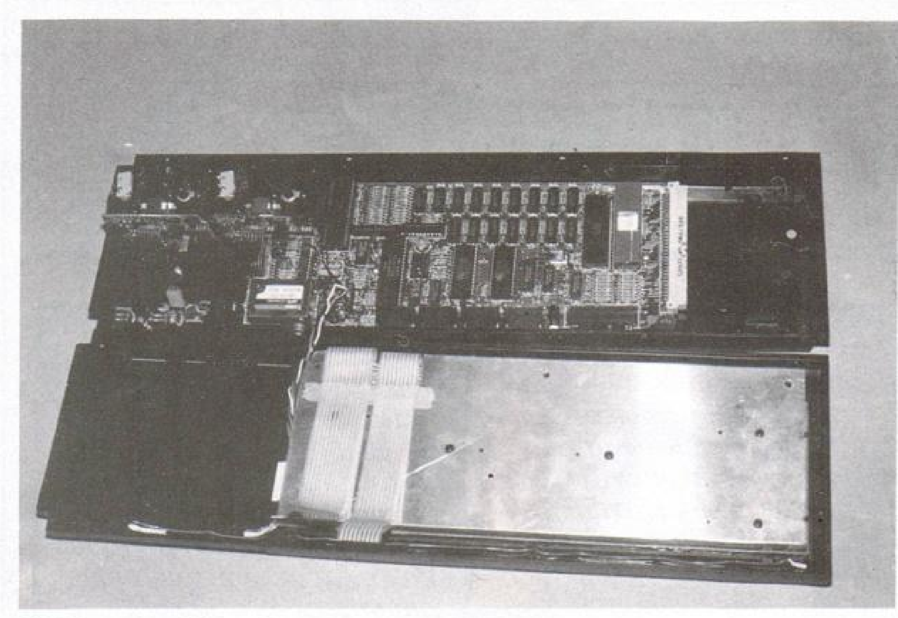
El 68008

En la parte izquierda de la placa se encuentra el microprocesador 68008. Este es el encargado de hacer que el QL funcione. En lugar de elegir uno de tantos microprocesadores de 8 *bits* (Z80, 6502, 6809) como otros fabricantes, Sinclair directamente saltó a uno de tercera generación. El 68008 es miembro de una familia de sofisticados procesadores de 16/32 *bits* que se presentaron en 1979. Estos están orientados hacia el programador, ya que poseen una estructura interna muy versátil. Posee 16 registros de propósito general de 32 *bits,* un contador de programa de 32 *bits* y un registro de estado de 16 *bits.* Soporta cinco tipos distintos de datos: *bits,* BDC (4 *bits*), *bytes,* palabras (16 *bits*) y palabras largas (32 *bits*). Además de esto existen 14 modos de direccionamiento distintos, que incluyen los de postincremento, predecremento, *offset* e indexado. Por último hay 56 instrucciones distintas con múltiples variantes, lo que le proporciona una flexibilidad asombrosa.

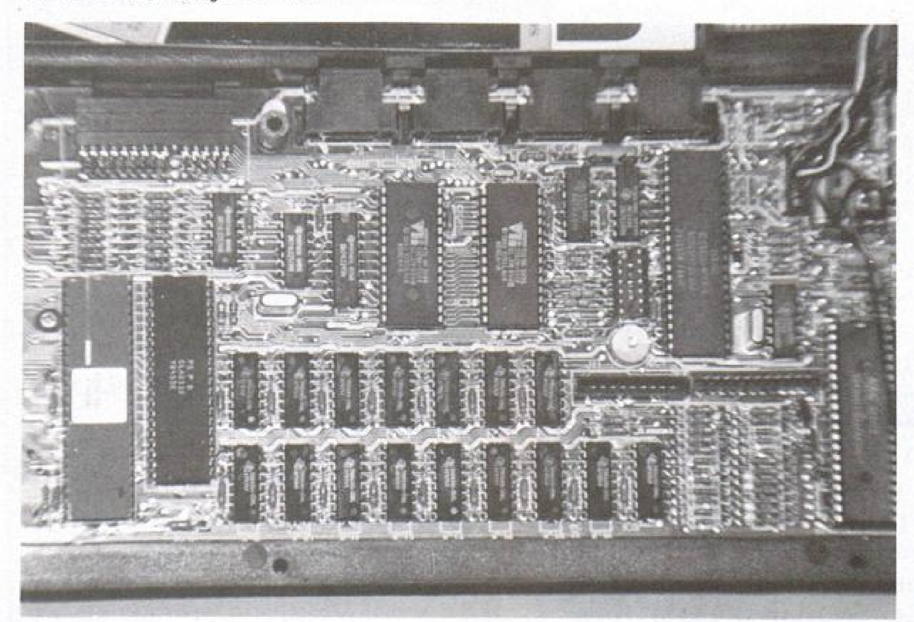
La única diferencia con su «hermano mayor», el 68000, es que el *bus* de datos del QL es de 8 *bits,* mientras que el de aquel es de 16, por lo que las instrucciones de lectura y escritura en memoria son más lentas. Pero como contrapartida se ha reducido costo en la memoria y en los circuitos encargados de controlar los periféricos.

Intel 8049

Este es otro microprocesador que lleva el ordenador y está situado en el lado opuesto de la placa. Se encarga de aliviarle trabajo al 68008 realizando tareas secundarias, tales como controlar el teclado, el generador de sonido y el puerto serie



El QL una vez abierto. Se puede ver en la parte inferior la placa metálica que cubre el teclado. En la parte superior está la placa principal, los *microdrives* (a su izquierda) y la fuente de alimentación debajo de éstos.



La placa principal del QL, en la cual se hallan los dos microprocesadores, la ROM y la RAM. El 68008 es el que tiene la placa dorada.

RS 232; dejando la CPU libre para ejecutar los programas del usuario. Su inclusión también ha sido necesaria para la correcta sincronización de todas aquellas tareas que requerían una medición precisa del tiempo, tarea que no es posible con el procesador principal, ya que éste está basado en interrupciones y éstas hacían imposible medir intervalos correctamente.

La ROM

Situada en la parte central trasera de la placa y formada por dos circuitos integrados con total de 32 *Kbytes,* contiene el sistema operativo y el SuperBasic.

La RAM

La memoria disponible para almacenamiento de datos, programas y los 32 K de la memoria de vídeo es de 128 K, repartidas en 16 circuitos situados delante de la ROM. Debido a que el 68008 es capaz de direccionar 1 *megabyte* de memoria, queda suficiente espacio de memoria para ampliar. Aunque todavía no ha habido anuncio oficial de Sinclair de un producto de este tipo, múltiples fabricantes lo ofrecen ya.

Conectores

El QL viene bien provisto de conectores para múltiples usos.

En la parte trasera existe gran cantidad de conectores. De izquierda a derecha nos encontramos con el de expansión de ROM, que permite meter un cartucho de 32 K. En los primeros modelos se usó como suplemento para incorporar parte del sistema operativo, ya que no cabía todo en la memoria principal. Actualmente está destinado para poder usar determinados programas que saldrán en formato de cartucho.

A continuación se encuentran los dos conectores de *joystick*, que no tienen un formato estándar, por lo que para conectarle un mando de este tipo habrá que hacer un cable especial. Seguidamente, y con un conector similar, se encuentran los dos del RS 232 que se destinan a controlar la impresora y para comunicaciones con otros ordenadores.

A su lado están el conector de televisión (del tipo habitual RCA) y un DIN de ocho puntas que da salida RGB para monitores. Finalmente a la derecha se encuentran los conectores de la red local.

Los microdrives

Estos dispositivos son la alternativa ofrecida por Sinclair al almacenamiento en *cassette* y ofrecen mucha mejor calidad que este sistema además de mayor velocidad. Como contrapartida hay que indicar que son más delicados de manejo. Si los comparamos con los *diskettes* de otros ordenadores, se ve que tienen menos capacidad (de 80 a 100 K, mientras los discos tienen de 250 a 800 K) y son más lentos (15 K de velocidad de transferencia), pero posee una contrapartida muy interesante que es el precio, ya que si se hubiesen incluido discos en la configuración básica el precio habría subido considerablemente.



El corcho

Su anuncio puede ir aquí. Escríbanos a «El corcho». TODOSPECTRUM. Bravo Murillo, 377. 5.ª A. 28020 Madrid

Vendo interface programable para joystick de Investrónica por 4.500 ptas. negociables. Preguntar por Antonio en el (91) 741 03 26.

¡S.O.S.! Vendo urgentemente ordenador Oric Atmos en perfecto estado, con revistas, trucos personales, joystick nuevo sin interface, un cassette con sus cables, juegos como Panic, Lone Raider, Defender, etc. Escribir a Luis García Arroyo. Maragall, 38. 3.°-3. Ripollet (Barcelona).

Compro el número 1 de *Todospectrum.* José Ramón Arauzo. Kalebarria, 2. 1.º-D. Durango (Bizkaia). Tel.: (94) 681 29 29

Vendo Sinclair ZX Spectrum 48 K con cables, transformador y manual en castellano. Buen estado. También regalo 150 juegos últimas novedades: West Bank, Exploading Fist, etc. Junto al ordenador también doy monitor profesional especial para ordenadores con todos sus accesorios. Todo por 40.000 ptas. Escribir o llamad a Rafael de Mingo Gil. Diana, 13. G. 28002 Madrid. Tel.: (91) 741 88 59.

Vendo lápiz óptico Dk'tronics completamente nuevo con cassette de demostración, interface y estuche original por sólo 4.500 ptas. Sin más gastos de envío. Escribir a José Antonio Rodríguez Ovalle. Apartado 28 Villafranca del Bierzo (León). Precio a discutir.

Vendo lote formado por 1 joystick más interface tipo Kempston, 65 programas de los mejores juegos y utilidades, 1 libro de programación Basic, revistas (1 Your Computer, 3 Micromanías, 5 Todospectrum, 19 Microhobby, 11 ZX). Todo por 7.000 ptas. contrareembolso. Escribir a Adolfo Velasco Crespo. Avda. de España, 5. Ponferrada (León).

Vendo ZX Spectrum 48 K en perfecto estado acompañdo por revistas y software valorados en más de 150.000 ptas. Todo por sólo 60.000 ptas. Escribir a: Carlos José Arnaiz. Ramón y Cajal, 1. Bajo. Miranda de Ebro (Burgos).

Vendo varios libros de informática y para el ZX Spectrum, como nuevos. Escribir a: Fernando Castán. Padre Manjón, 34. 5.º-E. 50010 Zaragoza.

Intercambio o vendo (muy baratos) programas para Spectrum 16 y 48 K. Escribir a: Fidel García. Torrecampo, 71. Vua de Córdoba (Córdoba).

Vendo ZX Spectrum 48 K, más 50 programas comerciales de reciente aparición (Exploding Fist, Herbets, Boxing, Dambusters, Hypersports...). Interesados llamar al tel.: (96) 351 82 76 de Valencia y preguntar por Antonio, de 19 a 22 horas.

Vendo ZX 81 más apliación de 16 K con manual en castellano, fuente de alimentación, programas (3D Monster Maze) por 10.000 ptas. Llamar a: Manuel Gregorio. Perú, 4. 1.º izda. Valladolid. Tel.: (983) 39 68 73 (llamar de 9 a 11 h.).

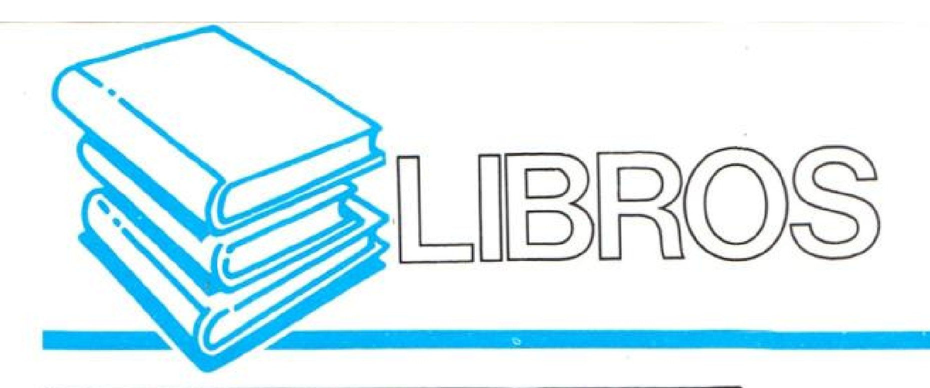
Vendo impresora Seikosha 50-S, no necesita interface. Llamar noche al tel.: (93) 783 55 39. Preguntar por Jordi Martínez. Precio a convenir. Tiene garantía.

Vendo ZX Spectrum 48 K en buenas condiciones con fuente de alimentación, cables, manual de instrucciones y 7 juegos, por 29.000 ptas. Interesados ponerse en contacto con: José María Ripa. Tel.: (91) 733 03 00. ext. 308. Madrid.

Vendo o cambio máquina de marcianos bastante buena y con diversas fases, se llama Galaxy II y la vendo por el sorpredente precio de 4.500 ptas. (en el mercado está entre 8.000 y 10.000 ptas.) o cambio por programa para Spectrum. Interesados llamar a: Daniel Olive Subias. Del Puente, 3. Vilavert (Tarragona). Tel.: (977) 87 64 53.





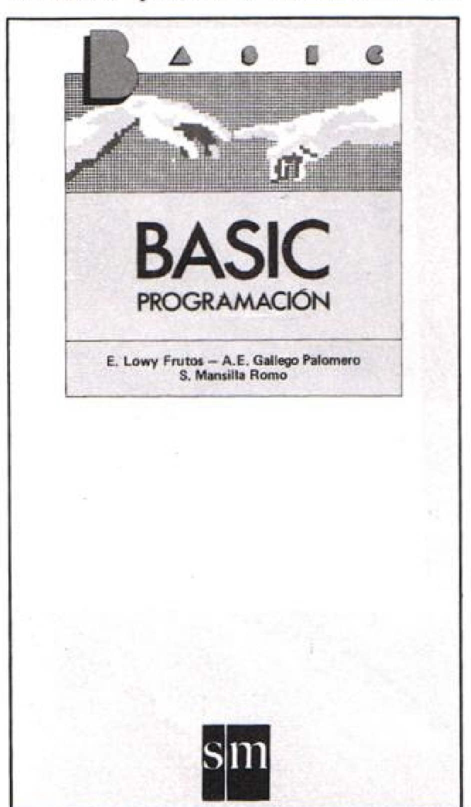


BASIC PROGRAMACION

Editorial: Ediciones SM
Autores: Lowy Frutos,
Gallego Palomero,
Mansilla Romo
Páginas 225.

Dentro del extenso número de libros publicados en los últimos años sobre informática destacan, sobre todo, los dedicados al aprendizaje del lenguaje BASIC y es, dentro de éste grupo, donde podemos incluir éste nuevo título, *BASIC Programación*, publicado por Ediciones SM y cuyos autores (dato a tener en cuenta) son un grupo de profesionales españoles.

Tanto la estructura como la distribución de las materias que abarca lo sitúan preferentemente como un libro didáctico, en el que cada capítulo o tema (consta de 14 capítulos y dos apéndices) va desglosando la estructura del BASIC, su funcionamiento, sentencias y aplicaciones prácticas ilustradas con



gran profusión de ejemplos y programas. Los capítulos tienen secciones fijas que incluyen la descripción y resumen de las sentencias, desarrollo de un programa de poca extensión y explicación de su funcionamiento y un buen número de ejercicios diversos para el usuario y de recapitulación, además de otros problemas resueltos por los autores de la forma más idónea.

En principio el libro no está redactado para un determinado sector de usuarios y máquinas. Su facilidad de comprensión lo hacen apto para cualquier usuario, tenga poca o mucha experiencia con el BASIC o con cualquier otro lenguaje, ya que tanto los ejercicios como la descripción y uso de las sentencias están suficientemente explicadas como para que no presenten ningún problema. Todos los programas y ejercicios pueden ejecutarse en cualquier microordenador, ya que se utilizan sentencias estándar del BASIC y, en aquellos casos en los que determinadas sentencias no sean aceptadas por algún ordenador, se especifica cuál y se establece la modificación oportuna. Aparte de esto, es aconsejable utilizar el manual del BASIC del ordenador en que se está trabajando junto con el libro de aprendizaje.

El libro incide fundamentalmente en las posibilidades matemáticas y de cálculo del BASIC, exponiendo e invitando al usuario con una gran variedad de ejemplos y ejercicios, para que utilice su máquina como una potente calculadora, sirviendo de este modo como un libro de apoyo al estudiante para repasar las lecciones aprendidas.

También resulta especialmente útil a los usuarios que pretenden aprender, de una forma clara y amena, a programar su ordenador para realizar sus propios programas de gestión, diversión, etcétera.

DISEÑO DE GRAFIC

Autor: I. O. Angel-B. J. Jones
Editorial: ANAYA Multimedia
Páginas: 387.

Resulta evidente, incluso para aquellas personas no introducidas en el mundo de la informática, que uno de los campos más apasionantes y vistosos de los ordenadores, es el dedicado a las representaciones gráficas en sus múltiples variantes. Una demostración clara de esto son los juegos que invaden todas las tiendas, en los que se puede ver un amplio desarrollo gráfico.

Dentro de esta corriente «gráfica» han surgido múltiples libros que intentan dar una explicación clara y sencilla de su manejo para que el lector pueda realizar sus propios programas. Este es el caso de «Diseño de gráficos y videojuegos», un libro que hace exhaustivo estudio de los gráficos del Spectrum en todas sus variantes. Esto mismo podría decirse sin duda de muchos otros, pero en este caso hay diferencias importantes. En primer lugar no se parte de la idea del recetario de trucos sencillos para usar en los programas, sino que se da la base matemática necesaria para su comprensión usando la geometría del plano y del espacio que muchos estudiantes habrán tenido que aprender. Esto hace que aunque en principio es más difícil de entender que otros libros, si se lee atentamente la comprensión del funcionamiento de los gráficos, y por tanto su manejo, es mucho más fácil.

El otro detalle que lo diferencia de otros libros es que incluye una cinta con algunos programas demostrativos y lo que es más importante, subrutinas avanzadas que se pueden introducir dentro de los programas que haga el lector.

El índice del libro consta de 15 capítulos y dos apéndices, lo que cubre prácticamente todo el espec-

OS Y VIDEOJUEGOS

tro de variantes gráficas del Spectrum.

A continuación se pasa al estudio de los caracteres gráficos con indicaciones para usar los ya existentes y para crear otros nuevos. Como aplicación de estos conceptos se dan listados de algunos juegos, como el mastermind y un sencillo ajedrez.

En el siguiente capítulo se habla de las aplicaciones estadísticas de los gráficos, realización de histogramas (gráficos de barras) y modo de calcularlos y representarlos.

Siguiendo con el tema de representación de objetos tridimensionales se ve posteriormente el tema de los algoritmos de líneas ocultas.



Estos son sistemas de representación de objetos en los que se calculan que partes no se ven debido a que están detrás de otras y en consecuencia no se dibujan.

Como final se incluye una parte destinada a los manejos avanzados de gráficos y sus aplicaciones. Entre estas destacan las de los videojuegos, a los que se dedica un capítulo entero explicando paso a paso su realización.

El libro es un interesante tratado del tema de los gráficos en todas sus variantes y resulta interesante para todas aquellas personas que quieran realizar una profundización seria de este tema. Cabe señalar también la cuidada traducción, realizada con un cuidado y esmero que debería servir de ejemplo a otros libros.







FIGURA 1

```
  10 CLEAR 65367: BORDER 4: PAPE
R 4: INK 4: CLS : POKE 23658,8:
POKE 65472,52: POKE 65473,74: PO
KE 23609,100
  15 LOAD ""CODE 65368,64: INK 9
: CLS
  20 PRINT  FLASH 1;AT 1,9;"INST
RUCCIONES!"
  30 PRINT ''"  JUEGAS AL MUS DE
 4 REYES"'"  EL JUEGO ES A 25 PI
EDRAS"'"  EN TOTAL 5 AMARRACOS"'
"  CONTESTA A LAS PREGUNTAS CON:
"
  40 PRINT '"  S  PARA CONTESTAR
 SI"''"  N  PARA NO"''"  E  PARA
 ENVIDO"''"  O  PARA ORDAGO"
  50 PRINT '"  SI QUIERES CAMBIA
R TODAS LAS    CARTAS PULSA ENTE
R"
  60 PRINT AT 21,3;"PULSA UNA TE
CLA PARA JUGAR": PAUSE 0: CLS
  80 REM CONDICION DE RANDOMIZE
  82 RANDOMIZE
 100 REM VARIABLES DEL PROGRAMA
 103 LET pie=0: LET piesp=0: LET
 amar=0: LET amarsp=0
```



JUGANDO AL MUS

¿Quién no ha visto alguna vez las frases «órdago», «envido», «paso» en alguno de los bares o tascas que pueblan la geografía española? El mus es, indudablemente, uno de los juegos con más raigambre nacional y por eso el programa cuyo listado damos y que nos ha sido remitido por Santiago Fernández Mayoral, es seguro que hará feliz a más de uno, ya que es capaz de jugar contra un oponente humano, realizando jugadas típicas y contando amarracos, aunque eso sí, no es capaz de hacer señas.

Hay que reconocer que el ordenador no juega muy bien, ya que es relativamente fácil ganarle; ahora bien, al estar escrito en BASIC cualquier osado puede tomar el listado del programa y mejorarlo para que sea más atrevido u osado.

El juego se compone de un listado principal que se da en la figura 1 y de una subrutina en lenguaje máquina que se crea en el listado de la figura 2 y que es la encargada de crear los caracteres gráficos. Para crear una cinta que funcione basta teclear el primer programa, salvarlo en cinta y a continuación teclear el segundo y ejecutarlo, con lo cual graba en la cinta, a continuación del programa principal, los códigos que posteriormente éste lee.

Hay que hacer notar que estos caracteres se muestran ya definidos en el listado, por lo que cada vez que aparezca uno, hay que introducir el caracter gráfico correspondiente según se muestra en la tabla que se incluye.

Para aquellos interesados en modificarlo damos la descripción de los bloques principales del programa:

Líneas 10 a 100: presentación del programa y *randomize*.

Líneas 100 a 130: adjudicación de variables del programa.

Línea 136: subrutina de cuenta de piedras y amarracos.

Línea 150: adjudicación de variables a la jugada.

Líneas 200 a 490: subrutina que da valores a las cartas y las dibuja.

Líneas 500 a 990: condiciones para que haya mus o no.

Líneas 1000 a 1990: jugada a grande con las diversas condiciones de envido, órdago, paso, etc.

Líneas 2000 a 2990: lo mismo a pequeña.

Líneas 3000 a 3990: lo mismo a pares.

Líneas 4000 a 4990: lo mismo a juego.

Líneas 5000 a 5990: cuenta final de puntos y vuelta a la línea 150.

Línea 6000: subrutina que dibuja las cartas de Spectrum.

Línea 7000: pregunta si hay nueva partida y despedida.

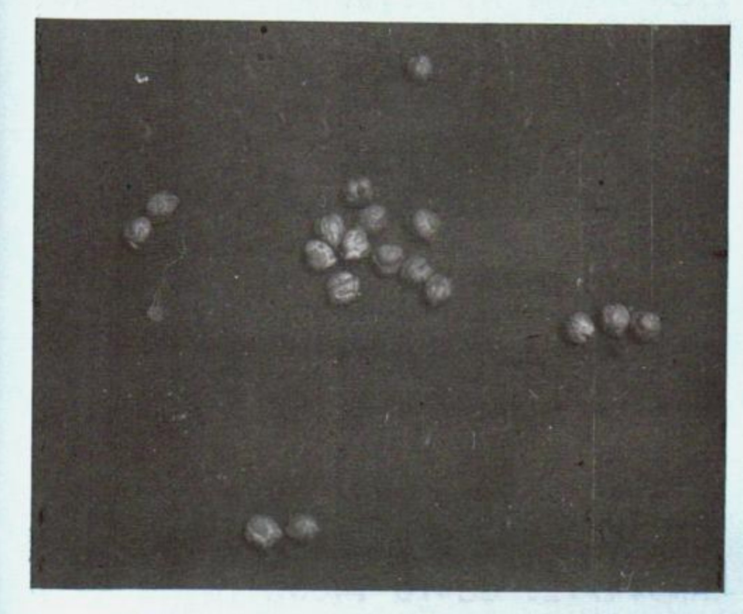
```
106 LET tot=0: LET totsp=0
109 LET sp=0
112 LET t=150
115 LET w=136
130 REM TABLERO CONTADOR
133 PRINT AT 7,12;"SPECTRUM";AT
14,12;" JUGADOR"
134 PRINT AT 18,0;"AMARRAC.";AT
18,25;"PIEDRAS";AT 3,0;"PIEDRAS
";AT 3,25;"AMARRAC"
135 GO SUB w: GO TO 150
137 IF pie>=20 THEN  LET amar=a
mar+4: LET pie=pie-20
138 IF pie>=15 AND pie<20 THEN
LET amar=amar+3: LET pie=pie-15
139 IF pie>=10 AND pie<15 THEN
LET amar=amar+2: LET pie=pie-10
140 IF pie>=5 AND pie<10 THEN
LET amar=amar+1: LET pie=pie-5
141 IF piesp>=20 THEN  LET amar
sp=amarsp+4: LET piesp=piesp-20
142 IF piesp>=15 AND piesp<20 T
HEN  LET amarsp=amarsp+3: LET pi
esp=piesp-15
143 IF piesp>=10 AND piesp<15 T
HEN  LET amarsp=amarsp+2: LET pi
```

```
esp=piesp-10
 144 IF piesp>=5 AND piesp<10 THEN  LET amarsp=amarsp+1: LET piesp=piesp-5
 145 LET tot=amar*5+pie
 146 LET totsp=amarsp*5+piesp
 148 PRINT AT 19,2;" ";amar;AT 19,27;" ";pie;AT 4,2;" ";piesp;AT 4,27;" ";amarsp
 149 BEEP .1,30: RETURN
 150 REM VARIABLES DE LA JUGADA
 153 DIM c$(36): DIM d$(36)
 156 LET e=0: LET x=0: LET l=0: LET cant=0: LET cantsp=0: LET mus=0: LET gran=0: LET gransp=0: LET peq=0: LET peqsp=0
 159 DIM f(8): DIM a(36): DIM b(36): DIM c(9): DIM e(8): DIM g(8): DIM p(8)
 162 LET c$="": LET d$=""
 165 LET nopa=0: LET noju=0
 168 LET no=0
 200 REM BARAJANDO
 206 GO SUB 212
 209 GO TO 500
 212 PRINT AT 11,0;"           REPARTO CARTAS           ": PAUSE t
 213 IF sp=0 THEN  PRINT  FLASH 1;AT 11,4;"          JUEGAS TU!      "
 214 IF sp=1 THEN  PRINT  FLASH 1;AT 11,5;"          JUEGO YO      "
 215 FOR n=1 TO 8
 217 IF f(n)=n THEN  GO TO 353
 218 LET a=INT (RND*9)+1
 219 LET b=INT (RND*4)+1
 220 FOR y=1 TO x
 221 IF a(y)=a AND b(y)=b THEN  GO TO 218
 222 NEXT y
 223 LET a(y)=a: LET b(y)=b
 224 LET x=x+1
 225 GO SUB 400
 230 LET c(n)=a
 233 LET e(n)=e
 234 LET g(n)=g
 235 LET p(n)=p
 236 REM DIBUJO DE CARTAS
 242 IF l>12 THEN  LET l=0
 245 IF n>4 THEN  GO TO 300
 250 REM DIBUJO CARTAS JUGADOR
 252 FOR m=16 TO 20
 253 PRINT  PAPER 7;AT m,9+l;"     ": NEXT m
 254 IF d$(n)="♆" THEN  LET ink=4
 255 IF d$(n)="↓" THEN  LET ink=1
 256 IF d$(n)="▒" THEN  LET ink=6
 257 IF d$(n)="↘" THEN  LET ink=2
 262 PRINT  PAPER 7; INK 9;AT 16,9+l;c$(n);AT 17,9+l; INK ink;d$(n)
 265 PRINT  PAPER 7; INK ink;AT 19,11+l;d$(n);AT 20,11+l; INK 9;c$(n)
```



```
 266 PLOT 71+l*8,48: DRAW 25,0: DRAW 0,-41: DRAW -25,0: DRAW 0,41
 268 GO TO 350
 300 REM DIBUJO CARTAS SPECTRUM
 303 FOR m=1 TO 5
 304 PRINT  PAPER 1; INK 7;AT m,9+l;"░░░░░"
 307 NEXT m
 319 PLOT 71+l*8,168: DRAW 25,0: DRAW 0,-41: DRAW -25,0: DRAW 0,41
 321 GO TO 350
 350 REM DIBUJAR LAS 8 CARTAS
 353 LET l=l+4
 354 BEEP .1,40
 356 NEXT n
 358 LET cant=e(1)+e(2)+e(3)+e(4)
```

```
 359 LET cantsp=e(5)+e(6)+e(7)+e
(8)
 360 LET gran=g(1)+g(2)+g(3)+g(4
)
 361 LET gransp=g(5)+g(6)+g(7)+g
(8)
 362 LET peq=p(1)+p(2)+p(3)+p(4)
 363 LET peqsp=p(5)+p(6)+p(7)+p(
8)
 365 RETURN
 400 REM VALORES CARTAS
 401 IF a=1 THEN  LET e=1: LET g
=1: LET p=6561: LET c$(n)="A"
 403 IF a=2 THEN  LET e=3: LET g
=3: LET p=2187: LET c$(n)="3"
 406 IF a=3 THEN  LET e=4: LET g
```



```
=9: LET p=729: LET c$(n)="4"
 409 IF a=4 THEN  LET e=5: LET g
=27: LET p=243: LET c$(n)="5"
 412 IF a=5 THEN  LET e=6: LET g
=81: LET p=81: LET c$(n)="6"
 415 IF a=6 THEN  LET e=7: LET g
=243: LET p=27: LET c$(n)="7"
 418 IF a=7 THEN  LET e=10: LET
g=729: LET p=9: LET c$(n)="10"
 421 IF a=8 THEN  LET e=10: LET
g=2187: LET p=3: LET c$(n)="11"
 424 IF a=9 THEN  LET e=10: LET
g=6561: LET p=1: LET c$(n)="12"
 427 IF b=1 THEN  LET d$(n)="Ψ"
 430 IF b=2 THEN  LET d$(n)="†"
 433 IF b=3 THEN  LET d$(n)="●"
 436 IF b=4 THEN  LET d$(n)="\"
 439 RETURN
 500 REM CONDICIONES DE MUS
 501 IF mus>1 THEN  GO TO 950
 502 DIM f(8)
 503 PRINT AT 11,0;"           TIE
NES MUS? S/N        "
 506 PAUSE 0
 509 IF INKEY$="N" THEN  GO TO 1
000
 512 IF INKEY$="S" THEN  GO TO 5
50
 515 IF INKEY$<>"S" AND INKEY$<>
"N" THEN  GO TO 503
 550 REM CONDICIONES MUS SP.
 553 IF cantsp>=30 THEN  GO TO 9
00
 556 IF (c(5)=9 AND c(6)=9) OR (
c(5)=9 AND c(7)=9) OR (c(5)=9 AN
D c(8)=9) OR (c(6)=9 AND c(7)=9)
 OR (c(6)=9 AND c(8)=9) OR (c(7)
=9 AND c(8)=9) THEN  GO TO 900
 559 IF (c(5)=8 AND c(6)=8) OR (
c(5)=8 AND c(7)=8) OR (c(5)=8 AN
D c(8)=8) OR (c(6)=8 AND c(7)=8)
 OR (c(6)=8 AND c(8)=8) OR (c(7)
=8 AND c(8)=8) THEN  GO TO 300
 562 IF (c(5)=c(6) AND c(5)=c(7)
) OR (c(5)=c(7) AND c(5)=c(8)) O
R (c(6)=c(7) AND c(5)=c(8)) THEN
  GO TO 900
 565 IF (c(5)=c(6) AND c(7)=c(8)
) OR (c(5)=c(7) AND c(6)=c(8)) O
R (c(5)=c(8) AND c(6)=c(7)) THEN
  GO TO 900
 568 PRINT AT 11,0;"        YO TAM
BIEN TENGO MUS        ": PAUSE t
 571 PRINT AT 11,0;"          CAMB
IO DE CARTAS          ": PAUSE t
 572 PRINT AT 11,0;"MARCA CARTA
"; FLASH 1;"A NO CAMBIAR"; FLASH
 0;" 1-2-3-4"
 575 PRINT AT 12,0;"      PARA AC
ABAR PULSA ENTER        "
 578 INPUT f$
 581 IF f$="" THEN  GO TO 600
 587 LET f=VAL f$
 590 LET f(f)=f
 593 GO TO 578
 600 REM SPECTRUM CAMBIA CARTAS
 603 IF c(5)>6 THEN  LET f(5)=5
 606 IF c(6)>6 THEN  LET f(6)=6
 609 IF c(7)>6 THEN  LET f(7)=7
 612 IF c(8)>6 THEN  LET f(8)=8
 615 LET mus=mus+1
 621 LET x=x+1
```

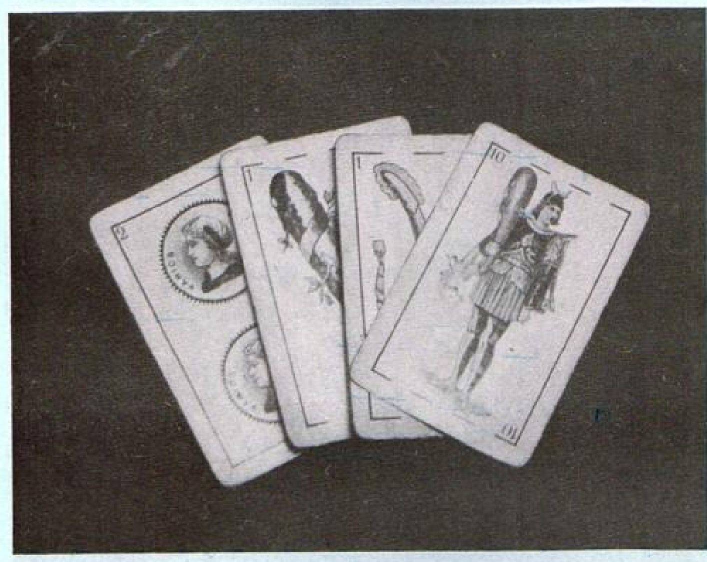
```
622 LET l=0
625 PRINT AT 11,0;"
                          "
630 GO SUB 212
633 GO TO 500
900 PRINT AT 11,0;"      PUES YO
NO, FASTIDIATE!   ": PAUSE t
903 GO TO 1000
950 PRINT AT 11,0;"      YA NO
HAY MAS MUS!    ": PAUSE t
953 GO TO 1000
1000 PRINT AT 11,0;"        JUGA
DA A GRANDE       ": PAUSE t
1003 LET gr=0: LET env=0: DIM j(
8): LET j=8: LET an=0
1004 IF sp=1 THEN  GO TO 1200
1006 PRINT AT 11,0;"    ENVIDAS
A GRANDE ? S/N/O   "
1009 REM LECTURA CARTAS SPECTRUM
1012 IF gransp>=17496 THEN  LET
j(j)=1
1015 IF gransp>=14580 AND gransp
<17496 THEN  LET j(j)=2
1018 IF gransp>=13124 AND gransp
<14580 THEN  LET j(j)=3
1021 IF gransp>=9478 AND gransp<
13124 THEN  LET j(j)=4
1023 IF gransp>=8850 AND gransp<
9478 THEN  LET j(j)=5
1025 IF gransp>=7292 AND gransp<
8850 THEN  LET j(j)=6
1027 PAUSE 0
1030 IF INKEY$="O" THEN  LET an=
0: GO TO 1300
1033 IF INKEY$="N" THEN  LET an=
1: LET no=1: GO TO 1200
1036 IF INKEY$="S" THEN  LET an=
0: GO TO 1042
1039 IF INKEY$<>"O" OR INKEY$<>"
N" OR INKEY$<>"S" THEN  GO TO 10
06
1042 PRINT AT 11,0;"
CUANTO?           ": INPUT z
1045 IF z<2 THEN  GO TO 1042
1046 LET env=env+1
1048 LET gr=gr+z
1051 IF j(j)=1 AND RND>.5 THEN
GO TO 1450
1054 IF j(j)=1 THEN  LET ap=INT
(RND*5)+2: GO TO 1400
1057 IF j(j)=2 AND RND>.75 THEN
 GO TO 1450
1060 IF j(j)=2 THEN  LET ap=INT
(RND*3)+2: GO TO 1400
1063 IF j(j)=3 AND gr<6 AND RND>
.85 THEN  GO TO 1450
1066 IF j(j)=3 AND gr<6 THEN  LE
T ap=INT (RND*1)+2: GO TO 1400
1069 IF j(j)=3 AND gr>6 THEN  GO
 TO 1100
1072 IF j(j)=4 AND gr<=4 THEN  G
O TO 1100
1073 IF RND<.15 THEN  LET ap=2:
GO TO 1400
1075 GO TO 1150
1100 PRINT AT 11,0;"
 QUIERO           ": PAUSE t:
GO TO 2000
1150 PRINT AT 11,0;"           N
O QUIERO          ": PAUSE t
1153 IF env=1 THEN  LET gr=0: LE
T pie=pie+1: GO SUB w
1156 GO TO 1990
1200 REM JUGADOR NO ENVIDA
1203 IF j(j)=1 AND RND>.4 THEN
GO TO 1450
1206 IF j(j)=1 THEN  LET ap=INT
(RND*8)+2: GO TO 1400
1209 IF j(j)=2 AND RND>.5 THEN
GO TO 1450
1212 IF j(j)=2 THEN  LET ap=INT
(RND*6)+2: GO TO 1400
1215 IF j(j)=3 AND RND>.75 THEN
 GO TO 1450
1218 IF j(j)=3 THEN  LET ap=INT
(RND*4)+2: GO TO 1400
1221 IF j(j)=4 AND RND>.9 THEN
GO TO 1450
1224 IF j(j)=4 THEN  LET ap=INT
(RND*3)+2: GO TO 1400
1227 IF j(j)=5 THEN  LET ap=INT
(RND*2)+2: GO TO 1400
1230 IF j(j)=6 THEN  LET ap=2: G
O TO 1400
1231 IF c(5)=9 OR c(6)=9 OR c(7)
=9 OR c(8)=9 THEN  LET ap=2: GO
TO 1400
1232 IF RND<2 THEN  LET ap=2: GO
 TO 1400
1233 PRINT AT 11,0;"        YO P
ASO A GRANDE      ": PAUSE t
1234 LET an=1
1236 IF sp=1 AND no=1 THEN  LET
no=0: GO TO 2000
1237 IF sp=1 THEN  GO TO 1006
```

```
1239 GO TO 1990
1300 REM ORDAGO
1303 PRINT AT 11,0;"        ME OR
DAGUEAS! EH!!         ": PAUSE t
1306 IF gransp>=13124 THEN  GO T
O 1350
1309 PRINT AT 11,0;"            N
O QUIERO!              ": PAUSE t
1310 IF env=0 THEN  LET pie=pie+
1: GO SUB w: LET gr=0: GO TO 199
0
1312 IF env>=1 THEN  LET pie=pie
+gr: LET gr=0: GO SUB w: GO TO 1
990
```



```
1350 PRINT AT 11,0;"
!QUIERO!               ": PAUSE t
1353 PRINT AT 11,0;"      VAMOS A
 VER LAS CARTAS        ": PAUSE t
1356 GO SUB 6000
1359 IF gransp>gran THEN  PRINT
AT 11,0;"           !!!!TE GANE!!!
      ": PAUSE t: GO TO 7000
1362 IF gransp<=gran THEN  PRINT
 AT 11,0;"         ME GANASTE
      ": PAUSE t: GO TO 7000
1400 REM SPECTRUM ENVIDA
1401 LET an=0
1402 LET env=env+1
1403 PRINT AT 11,0;"YO TE ENVIDO
 ";ap;" QUIERES S/N/E/O": PAUSE
0
1406 LET gr=gr+ap
1409 IF INKEY$="S" THEN  GO TO 2
000
1410 IF INKEY$="N" AND env=1 THE
N  LET gr=0: LET piesp=piesp+1:
GO SUB w: GO TO 1990
1412 IF INKEY$="N" AND env>1 THE
N  LET gr=gr-ap: LET piesp=piesp
+gr: GO SUB w: LET gr=0: GO TO 1
990
1415 IF INKEY$="O" THEN  GO TO 1
300
1418 IF INKEY$="E" THEN  LET env
=env+1: GO TO 1424
1421 IF INKEY$<>"S" OR INKEY$<>"
N" OR INKEY$<>"O" THEN  GO TO 14
03
1424 PRINT AT 11,0;"
 CUANTO?               ": INPUT z
1427 IF z<2 THEN  GO TO 1424
1430 IF z<INT (RND*10)+2 THEN  G
O TO 1442
1433 PRINT AT 11,0;"ES DEMASIADO
 PARA MI, NO QUIERO ": PAUSE t:
LET pie=pie+gr: GO SUB w: LET gr
=0: PAUSE t: GO TO 1990
1442 PRINT AT 11,0;"
!QUIERO!               ": LET gr=gr
+z: PAUSE t: GO TO 2000
1450 REM ORDAGO DE SPECTRUM
1451 LET an=0
1453 PRINT AT 11,0; FLASH 1;"
  !ORDAGO! QUIERES S/N           ":
PAUSE 0
1456 IF INKEY$="N" AND env=1 THE
N  LET piesp=piesp+gr: GO SUB w:
 LET gr=0: GO TO 1990
1459 IF INKEY$="N" THEN  LET pie
sp=piesp+1: GO SUB w: LET gr=0:
GO TO 1990
1462 IF INKEY$="S" THEN  GO TO 1
468
1465 IF INKEY$<>"S" OR INKEY$<>"
N" THEN  GO TO 1453
1468 GO TO 1353
1990 IF totsp>=25 THEN  GO TO 65
00
1995 IF tot>=25 THEN  GO TO 6600
2000 PRINT AT 11,0;"          JUGA
DA A PEQUEÑA           ": PAUSE t
2003 LET eq=0: LET pe=0: LET env
=0: DIM k(8): LET k=8
2004 IF sp=1 THEN  GO TO 2200
2006 PRINT AT 11,0;"   ENVIDAS A
 PEQUEÑA? S/N/O        "
2009 REM LECTURA CARTAS SPECTRUM
```

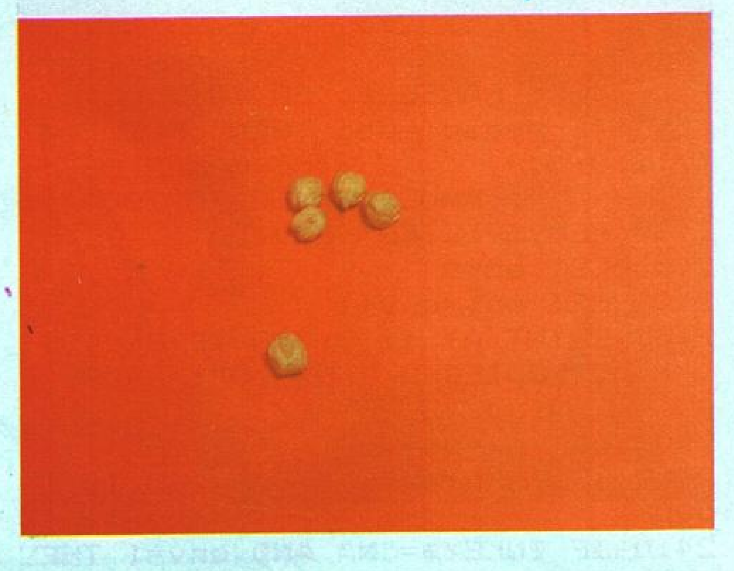
```
2012 IF peqsp>=17496 THEN  LET k
(k)=1
2015 IF peqsp>=14580 AND peqsp<1
7496 THEN  LET k(k)=2
2018 IF peqsp>=13124 AND peqsp<1
4580 THEN  LET k(k)=3
2021 IF peqsp>=9478 AND peqsp<13
124 THEN  LET k(k)=4
2023 IF peqsp>=8850 AND peqsp<94
78 THEN  LET k(k)=5
2025 IF peqsp>=7292 AND peqsp<88
50 THEN LET k(k)=6
2027 PAUSE 0
2030 IF INKEY$="O" THEN  LET eq=
0: GO TO 2300
2033 IF INKEY$="N" THEN  LET eq=
1: LET no=1: GO TO 2200
2036 IF INKEY$="S" THEN  LET eq=
0: GO TO 2042
2039 IF INKEY$<>"O" OR INKEY$<>"
N" OR INKEY$<>"S" THEN  GO TO 20
06
2042 PRINT AT 11,0;"
CUANTO?           ": INPUT z
2045 IF z<2 THEN  GO TO 2042
2046 LET env=env+1
2048 LET pe=pe+z
2051 IF k(k)=1 AND RND>.5 THEN
GO TO 2450
2054 IF k(k)=1 THEN  LET ap=INT
(RND*5)+2: GO TO 2400
2057 IF k(k)=2 AND RND>.75 THEN
 GO TO 2450
2060 IF k(k)=2 THEN  LET ap=INT
(RND*3)+2: GO TO 2400
2063 IF k(k)=3 AND pe<6 AND RND>
.85 THEN  GO TO 2450
2066 IF k(k)=3 AND pe<6 THEN  LE
T ap=INT (RND*1)+2: GO TO 2400
2069 IF k(k)=3 AND pe>6 THEN  GO
 TO 2100
2072 IF k(k)=4 AND pe<=4 THEN  G
O TO 2100
2073 IF RND<.15 THEN  LET ap=2:
GO TO 2400
2075 GO TO 2150
2100 PRINT AT 11,0;"
 QUIERO          ": PAUSE t:
GO TO 3000
2150 PRINT AT 11,0;"            N
O QUIERO         ": PAUSE t
2153 IF env=1 THEN  LET pe=0: LE
T pie=pie+1: GO SUB w
2156 GO TO 2990
2200 REM JUGADOR NO ENVIDA
2203 IF k(k)=1 AND RND>.4 THEN
GO TO 2450
2206 IF k(k)=1 THEN  LET ap=INT
(RND*8)+2: GO TO 2400
2209 IF k(k)=2 AND RND>.5 THEN
GO TO 2450
2212 IF k(k)=2 THEN  LET ap=INT
(RND*6)+2: GO TO 2400
2215 IF k(k)=3 AND RND>.75 THEN
 GO TO 2450
2218 IF k(k)=3 THEN  LET ap=INT
(RND*4)+2: GO TO 2400
```



```
2221 IF k(k)=4 AND RND>.9 THEN
GO TO 2450
2224 IF k(k)=4 THEN  LET ap=INT
(RND*3)+2: GO TO 2400
2227 IF k(k)=5 THEN  LET ap=INT
(RND*2)+2: GO TO 2400
2230 IF k(k)=6 THEN  LET ap=2: G
O TO 2400
2231 IF c(5)=1 OR c(6)=1 OR c(7)
=1 OR c(8)=1 THEN  LET ap=2: GO
TO 2400
2232 IF RND<.2 THEN  LET ap=2: G
O TO 2400
2233 PRINT AT 11,0;"          YO P
ASO A PEQUEÑA         ": PAUSE t
2234 LET eq=1:
2236 IF sp=1 AND no=1 THEN  LET
no=0: GO TO 3000
2237 IF sp=1 THEN  GO TO 2006
2239 GO TO 2990
2300 REM ORDAGO
2303 PRINT AT 11,0;"        ME OR
```

```
DAGUEAS! EH!!       ": PAUSE t
2306 IF peqsp>=13124 THEN  GO TO
 2350
2309 PRINT AT 11,0;"           N
O QUIERO!       ": PAUSE t
2310 IF env=0 THEN  LET pie=pie+
1: GO SUB w: LET pe=0: GO TO 299
0
2312 IF env>=1 THEN  LET pie=pie
+pe: GO SUB w: LET pe=0: GO TO 2
990
2350 PRINT AT 11,0;"
!QUIERO!            ": PAUSE t
2353 PRINT AT 11,0;"     VAMOS A
 VER LAS CARTAS       ": PAUSE t
2356 GO SUB 6000
2359 IF peqsp>peq THEN  PRINT AT
 11,0;"      !!!!TE GANE!!!
     ": PAUSE t: GO TO 7000
2362 IF peqsp<=peq THEN  PRINT A
T 11,0;"          ME GANASTE
     ": PAUSE t: GO TO 7000
2400 REM SPECTRUM ENVIDA
2401 LET eq=0
2402 LET env=env+1
2403 PRINT AT 11,0;"YO TE ENVIDO
 ";ap;" QUIERES S/N/E/O": PAUSE
0
2406 LET pe=pe+ap
2409 IF INKEY$="S" THEN  GO TO 3
000
2410 IF INKEY$="N" AND env=1 THE
N  LET pe=0: LET piesp=piesp+1:
GO SUB w: GO TO 2990
2412 IF INKEY$="N" AND env>1 THE
N  LET pe=pe-ap: LET piesp=piesp
+pe: GO SUB w: LET pe=0: GO TO 2
990
2415 IF INKEY$="O" THEN  GO TO 2
300
2418 IF INKEY$="E" THEN  LET env
=env+1: GO TO 2424
2421 IF INKEY$<>"S" OR INKEY$<>"
N" OR INKEY$<>"O" THEN  GO TO 24
03
2424 PRINT AT 11,0;"
 CUANTO?            ": INPUT z
2427 IF z<2 THEN  GO TO 2424
2430 IF z<INT (RND*10)+2 THEN  G
O TO 2442
2433 PRINT AT 11,0;"ES DEMASIADO
 PARA MI, NO QUIERO ": PAUSE t:
LET pie=pie+pe: GO SUB w: LET pe
=0: PAUSE t: GO TO 2990
2442 PRINT AT 11,0;"
!QUIERO!            ": LET pe=pe
+z: PAUSE t: GO TO 3000
2450 REM ORDAGO DE SPECTRUM
2451 LET eq=0
2453 PRINT AT 11,0; FLASH 1;"
     !ORDAGO! QUIERES S/N       ":
PAUSE 0
2456 IF INKEY$="N" AND env=1 THE
N  LET piesp=piesp+pe: GO SUB w:
 LET pe=0: GO TO 2990
2459 IF INKEY$="N" THEN  LET pie
sp=piesp+1: GO SUB w: LET pe=0:
GO TO 2990
2462 IF INKEY$="S" THEN  GO TO 2
468
2465 IF INKEY$<>"S" OR INKEY$<>"
N" THEN  GO TO 2453
2468 GO TO 2353
2990 IF totsp>=25 THEN  GO TO 65
00
2995 IF tot>=25 THEN  GO TO 6600
3000 PRINT AT 11,0;"         JUG
ADA A PARES          ": PAUSE t
3003 LET pa=0: LET env=0: DIM o(
3): LET o=2: DIM q(3): LET q=3:
LET res=0: LET resp=0
3004 LET er=0: LET ersp=0
3006 REM LECTURA PARES JUGADOR
          PARA VISION AL RECUENTO
3009 IF c(1)=c(2) OR c(1)=c(3) O
R c(1)=c(4) THEN  LET o(o)=1: LE
T er=2*c(1)
3010 IF c(2)=c(3) OR c(2)=c(4) T
HEN  LET o(o)=1: LET er=2*c(2)
3011 IF c(3)=c(4) THEN  LET o(o)
=1: LET er=2*c(3)
3012 IF (c(1)=c(2) AND c(1)=c(3)
) OR (c(1)=c(2) AND c(1)=c(4)) T
HEN  LET o(o)=2: LET er=3*c(1)
3013 IF c(2)=c(3) AND c(2)=c(4)
THEN  LET o(o)=2: LET er=3*c(2)
3015 IF (c(1)=c(2) AND c(3)=c(4)
) OR (c(1)=c(3) AND c(2)=c(4)) O
R (c(1)=c(4) AND c(2)=c(3)) THEN
  LET o(o)=3
3018 REM LECTURA PARES SPECTRUM
3021 IF c(5)=c(6) OR c(5)=c(7) O
R c(5)=c(8) THEN  LET q(q)=1: LE
T ersp=2*c(5)
3022 IF c(6)=c(7) OR c(6)=c(8) T
HEN  LET q(q)=1: LET ersp=2*c(6)
```

```
3023 IF c(7)=c(8) THEN LET q(q)
=1: LET ersp=2*c(7)
3024 IF (c(5)=c(6) AND c(5)=c(7)
) OR (c(5)=c(6) AND c(5)=c(8)) T
HEN LET q(q)=2: LET ersp=3*c(5)
3025 IF (c(6)=c(7) AND c(6)=c(8)
) THEN LET q(q)=2: LET ersp=3*c
(6)
3027 IF (c(5)=c(6) AND c(7)=c(8)
) OR (c(5)=c(7) AND c(6)=c(8)) O
R (c(5)=c(8) AND c(6)=c(7)) THEN
 LET q(q)=3
3030 REM COMIENZA JUEGO PARES
3033 PRINT AT 11,0;"          TIEN
ES PARES? S/N          ": PAUSE 0
3036 IF INKEY$="N" AND (q(q)=1 O
R q(q)=2 OR q(q)=3) THEN LET re
sp=1: GO TO 3800
3039 IF INKEY$="S" AND (q(q)=1 O
R q(q)=2 OR q(q)=3) THEN PAUSE
150: GO TO 3051
3042 IF INKEY$="S" THEN LET res
=1: GO TO 3900
3045 IF INKEY$="N" THEN GO TO 3
950
3048 IF INKEY$<>"S" OR INKEY$<>"
N" THEN GO TO 3033
3051 PRINT AT 11,0;"      YO TAMB
IEN TENGO PARES      ": PAUSE t
3052 IF sp=1 THEN GO TO 3200
3053 PRINT AT 11,0;"           ENV
IDAS? S/N/O          ": PAUSE 0
3054 IF INKEY$="N" THEN LET no=
1: GO TO 3200
3055 IF INKEY$="O" THEN GO TO 3
300
3057 IF INKEY$="S" THEN GO TO 3
061
3059 IF INKEY$<>"N" OR INKEY$<>"
O" OR INKEY$<>"S" THEN GO TO 30
53
3061 PRINT AT 11,0;"
 CUANTO?             ": INPUT z
3063 IF z<2 THEN GO TO 3061
3064 LET env=env+1
3065 LET pa=pa+z
3067 IF q(q)=3 AND gransp>=13124
 THEN GO TO 3450
3069 IF q(q)=3 THEN LET ap=INT
(RND*8)+2: GO TO 3400
3071 IF q(q)=2 AND gransp>=2187
THEN GO TO 3450
3073 IF q(q)=2 AND RND>.75 THEN
 GO TO 3450
3075 IF q(q)=2 THEN LET ap=INT
(RND*4)+2: GO TO 3400
3077 IF q(q)=1 AND gransp>=13122
 AND RND>.85 THEN GO TO 3450
3079 IF q(q)=1 AND gransp>=13122
 THEN LET ap=2: GO TO 3400
3081 IF q(q)=1 AND gransp>=4374
THEN GO TO 3100
3082 IF RND<.15 THEN LET ap=2:
GO TO 3400
3083 GO TO 3150
3100 PRINT AT 11,0;"
 QUIERO                 ": PAUSE t:
GO TO 4000
```

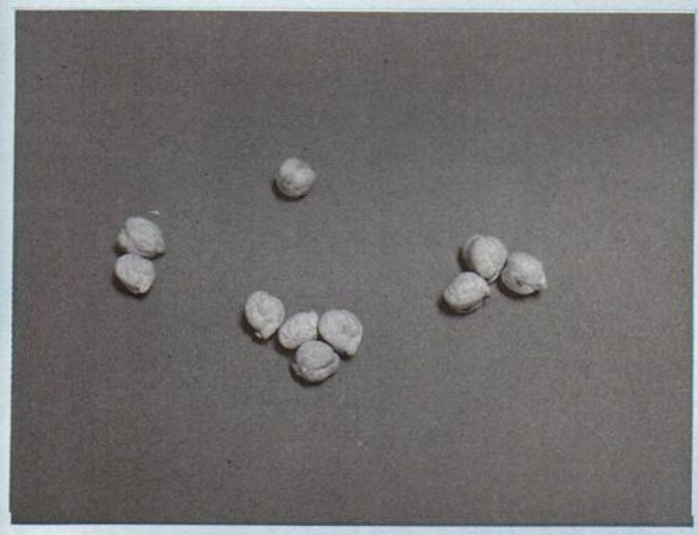


```
3150 PRINT AT 11,0;"             N
O QUIERO             ": PAUSE t
3153 IF env=1 THEN LET pa=0: LE
T pie=pie+1: GO SUB w
3154 LET res=1
3156 GO TO 3990
3200 REM JUGADOR NO ENVIDA
3203 IF q(q)=3 AND gransp>=13124
 THEN GO TO 3450
3206 IF q(q)=3 AND gransp>=4374
AND RND>.3 THEN GO TO 3450
3209 IF q(q)=3 AND gransp>=1458
AND RND>.5 THEN GO TO 3450
3212 IF q(q)=3 AND gransp>=486 A
ND RND>.7 THEN GO TO 3450
3215 IF q(q)=3 AND RND>.85 THEN
 GO TO 3450
3218 IF q(q)=3 THEN LET ap=INT
(RND*10)+2: GO TO 3400
3221 IF q(q)=2 AND gransp>=19683
 AND RND>.5 THEN GO TO 3450
```

```
3224 IF q(q)=2 AND gransp>=19683
 THEN LET ap=INT (RND*6)+2: GO
TO 3400
3227 IF q(q)=2 AND gransp>=6561
AND RND>.7 THEN  GO TO 3450
3230 IF q(q)=2 AND gransp>=6561
THEN LET ap=INT (RND*4)+2: GO T
O 3400
3233 IF q(q)=2 AND gransp>=2187
AND RND>.9 THEN GO TO 3450
3236 IF q(q)=2 AND gransp>=2187
THEN LET ap=INT (RND*3)+2: GO T
O 3400
3239 IF q(q)=2 THEN  LET ap=INT
(RND*2)+2: GO TO 3400
3242 IF q(q)=1 AND gransp>=13122
 AND RND>.6 THEN  GO TO 3450
3245 IF q(q)=1 AND gransp>=13122
 THEN  LET ap=INT (RND*4)+2: GO
TO 3400
3248 IF q(q)=1 AND gransp>=4374
AND RND>.85 THEN  GO TO 3450
3251 IF q(q)=1 AND gransp>=4374
THEN  LET ap=INT (RND*2)+2: GO T
O 3400
3254 IF q(q)=1 AND gransp>=1458
AND RND>.95 THEN  GO TO 3450
3257 IF q(q)=1 AND gransp>=1458
THEN  LET ap=INT (RND*1)+2: GO T
O 3400
3260 IF RND<.2 THEN  LET ap=2: G
O TO 3400
3280 PRINT AT 11,0;"          YO
PASO A PARES        ": PAUSE t
3283 IF sp=1 AND no=1 THEN  LET
no=0: GO TO 4000
3284 IF sp=1 THEN  GO TO 3053
3286 GO TO 3990
3300 REM ORDAGO
3303 PRINT AT 11,0;"        ME OR
DAGUEAS! EH!!       ": PAUSE t
3306 IF q(q)=3 THEN  GO TO 3350
3309 IF q(q)=2 AND RND>.4 THEN
GO TO 3350
3310 IF q(q)=1 AND gransp>=13121
 THEN  GO TO 3350
3312 PRINT AT 11,0;"            N
O QUIERO!         ": PAUSE t
3313 LET res=1
3314 IF env=0 THEN  LET pie=pie+
1: GO SUB w: LET pa=0: GO TO 399
0
3315 IF env>=1 THEN  LET pie=pie
+pa: GO SUB w: LET pa=0: GO TO 3
990
3350 PRINT AT 11,0;"
!QUIERO!             ": PAUSE t
3353 PRINT AT 11,0;"        VAMOS A
 VER LAS CARTAS       ": PAUSE t
3356 GO SUB 6000
3359 IF q(q)=3 AND (o(o)=2 OR o(
o)=1) THEN  GO TO 3380
3361 IF q(q)=2 AND o(o)=1 THEN
GO TO 3380
3363 IF q(q)=3 AND o(o)=3 AND gr
ansp>gran THEN  GO TO 3380
3365 IF q(q)=2 AND o(o)=2 AND gr
ansp>gran THEN  GO TO 3380
3367 IF q(q)=1 AND o(o)=1 AND gr
ansp>gran THEN  GO TO 3380
3369 PRINT AT 11,0;"            ME
 HAS GANADO!          ": PAUSE t:
GO TO 7000
3380 PRINT AT 11,0;"            TE
 HE GANADO            ": PAUSE t:
GO TO 7000
3400 REM SPECTRUM ENVIDA
3402 LET env=env+1
3403 PRINT AT 11,0;"YO TE ENVIDO
 ";ap;" QUIERES S/N/E/O": PAUSE
0
3406 LET pa=pa+ap
3409 IF INKEY$="S" THEN  GO TO 4
000
3410 IF INKEY$="N" AND env=1 THE
N  LET pa=0: LET piesp=piesp+1:
GO SUB w: LET resp=1: GO TO 3990
3412 IF INKEY$="N" AND env>1 THE
N  LET pa=pa-ap: LET piesp=piesp
+pa: LET pa=0: LET resp=1: GO SU
B w: GO TO 3990
3415 IF INKEY$="O" THEN  GO TO 3
300
3418 IF INKEY$="E" THEN  LET env
=env+1: GO TO 3424
3421 IF INKEY$<>"S" OR INKEY$<>"
N" OR INKEY$<>"O" THEN  GO TO 34
03
3424 PRINT AT 11,0;"
 CUANTO?                ": INPUT z
3427 IF z<2 THEN  GO TO 3424
3430 IF z<INT (RND*10)+2 THEN  G
O TO 3442
3433 PRINT AT 11,0;"ES DEMASIADO
 PARA MI, NO QUIERO ": PAUSE t:
LET pie=pie+pa: GO SUB w: LET pa
```

```
=0: LET res=1: PAUSE t: GO TO 39
90
3442 PRINT AT 11,0;"
!QUIERO!            ": LET pa=pa
+z: PAUSE t: GO TO 4000
3450 REM ORDAGO DE SPECTRUM
3453 PRINT AT 11,0; FLASH 1;"
   !ORDAGO! QUIERES S/N       ":
PAUSE 0
3456 IF INKEY$="N" AND env=1 THE
N  LET piesp=piesp+pa: LET resp=
1: GO SUB w: LET pa=0: GO TO 399
0
3459 IF INKEY$="N" THEN  LET pie
sp=piesp+1: GO SUB w: LET pa=0:
LET resp=1: GO TO 3990
3462 IF INKEY$="S" THEN  GO TO 3
468
3465 IF INKEY$<>"S" OR INKEY$<>"
N" THEN  GO TO 3453
3468 GO TO 3353
3800 PRINT AT 11,0;"         YO
TENGO PARES            ": PAUSE t:
GO TO 4000
3900 PRINT AT 11,0;"         YO N
O TENGO PARES         ": PAUSE t:
GO TO 4000
3950 PRINT AT 11,0;"         YO N
O TENGO PARES      ": LET nopa=
1: PAUSE t: GO TO 4000
3990 IF totsp>=25 THEN  GO TO 65
00
3995 IF tot>=25 THEN  GO TO 6600
4000 PRINT AT 11,0;"          JUG
ADA A JUEGO          ": PAUSE t
4003 LET ju=0: LET env=0: DIM r(
8): LET r=8: LET ego=0: LET egos
p=0
4006 REM LECTURA JUEGO SPECTRUM
4008 IF cantsp=31 THEN  LET r(r)
=1
4010 IF cantsp=32 THEN  LET r(r)
=2
4012 IF cantsp=40 THEN  LET r(r)
=3
4014 IF cantsp=37 THEN  LET r(r)
=4
4016 IF cantsp=36 THEN  LET r(r)
=5
4018 IF cantsp=35 THEN  LET r(r)
=6
4020 IF cantsp=34 THEN  LET r(r)
=7
4022 IF cantsp=33 THEN  LET r(r)
=8
4030 REM COMIENZA EL JUEGO
4033 PRINT AT 11,0;"        TIENE
S JUEGO? S/N        ": PAUSE 0
4036 IF INKEY$="N" AND cantsp>=3
1 THEN  LET egosp=1: GO TO 4800
4039 IF INKEY$="N" AND cantsp<31
 THEN  GO TO 4950
4042 IF INKEY$="S" AND cantsp<31
 THEN  LET ego=1: GO TO 4900
4045 IF INKEY$="S" AND cantsp>=3
1 THEN  GO TO 4051
4048 IF INKEY$<>"S" OR INKEY$<>"
N" THEN  GO TO 4033
4051 PRINT AT 11,0;"       YO TAMB
IEN TENGO JUEGO      ": PAUSE t
4052 IF sp=1 THEN  GO TO 4200
4053 PRINT AT 11,0;"           ENV
IDAS? S/N/O          ": PAUSE 0
4054 IF INKEY$="N" THEN  LET no=
1: GO TO 4200
4055 IF INKEY$="O" THEN  GO TO 4
300
4057 IF INKEY$="S" THEN  GO TO 4
061
4059 IF INKEY$<>"S" OR INKEY$<>"
O" OR INKEY$<>"N" THEN  GO TO 40
53
4061 PRINT AT 11,0;"
 CUANTO?              ": INPUT z
4063 IF z<2 THEN  GO TO 4061
4064 LET env=env+1
4065 LET ju=ju+z
4067 IF r(r)=1 AND RND>.3 THEN
GO TO 4450
4069 IF r(r)=1 THEN  LET ap=INT
(RND*10)+2: GO TO 4400
4071 IF r(r)=2 AND RND>.6 THEN
GO TO 4450
4073 IF r(r)=2 THEN  LET ap=INT
(RND*6)+2: GO TO 4400
4075 IF r(r)=3 THEN  LET ap=INT
(RND*4)+2: GO TO 4400
4077 IF r(r)=4 THEN  LET ap=INT
(RND*2)+2: GO TO 4400
4079 IF RND<.15 THEN  LET ap=2:
GO TO 4400
4081 IF RND<.7 THEN  GO TO 4150
4100 PRINT AT 11,0;"
 QUIERO              ": PAUSE t:
GO TO 5500
4150 PRINT AT 11,0;"             N
```

```
O QUIERO!          ": PAUSE t
4151 LET ego=1
4153 IF env=1 THEN  LET ju=0: LE
T pie=pie+1: GO SUB w: GO TO 499
0
4200 REM JUGADOR NO ENVIDA
4203 IF r(r)=1 AND RND>.2 THEN
GO TO 4450
4206 IF r(r)=1 THEN  LET ap=INT
(RND*10)+2: GO TO 4400
4209 IF r(r)=2 AND RND>.5 THEN
GO TO 4450
4212 IF r(r)=2 THEN  LET ap=INT
(RND*8)+2: GO TO 4400
```



```
4215 IF r(r)=3 AND RND>.75 THEN
 GO TO 4450
4218 IF r(r)=3 THEN  LET ap=INT
(RND*6)+2: GO TO 4400
4221 IF r(r)=4 AND RND>.9 THEN
GO TO 4450
4224 IF r(r)=4 THEN  LET ap=INT
(RND*4)+2: GO TO 4400
4227 IF r(r)=5 THEN  LET ap=INT
(RND*2)+2: GO TO 4400
4230 IF r(r)=6 THEN  LET ap=2: G
O TO 4400
4233 IF r(r)=8 AND RND>.95 THEN
 GO TO 4450
4236 IF RND<.2 THEN  LET ap=2: G
O TO 4400
4239 PRINT AT 11,0;"          YO
PASO A JUEGO       ": PAUSE t
4242 IF sp=1 AND no=1 THEN  LET
no=0: GO TO 5500
4243 IF sp=1 THEN  GO TO 4053
4245 GO TO 4990
4300 REM ORDAGO
4303 PRINT AT 11,0;"          ME
 ORDAGUEAS! EH?     ": PAUSE t
4306 IF r(r)=1 AND RND>.4 THEN
GO TO 4350
4309 IF r(r)=2 AND RND>.6 THEN
GO TO 4350
4312 IF r(r)=3 AND RND>.75 THEN
 GO TO 4350
4315 IF r(r)=4 AND RND>.9 THEN
GO TO 4350
4318 PRINT AT 11,0;"
NO QUIERO!         ": PAUSE t
4319 LET ego=1
4320 IF env=0 THEN  LET pie=pie+
1: GO SUB w: LET ju=0: GO TO 499
0
4321 IF env>=1 THEN  LET pie=pie
+ju: GO SUB w: LET ju=0: GO TO 4
990
4350 PRINT AT 11,0;"
!!QUIERO!!         ": PAUSE t
4353 PRINT AT 11,0;"      VAMOS A
 VER LAS CARTAS     ": PAUSE t:
GO SUB 6000
4356 IF cantsp=31 AND cant<>31 T
HEN  GO TO 4370
4357 IF cantsp=32 AND (cant<>31
OR cant=32) THEN  GO TO 4370
4358 IF cantsp>cant THEN  GO TO
4370
4359 PRINT AT 11,0;"          ME
HAS GANADO!        ": PAUSE t:
GO TO 7000
4370 PRINT AT 11,0;"        !!TE
 HE GANADO!!        ": PAUSE t:
GO TO 7000
4400 REM SPECTRUM ENVIDA
4402 LET env=env+1
4403 PRINT AT 11,0;"YO TE ENVIDO
";ap;" QUIERES S/N/E/O": PAUSE
0
4406 LET ju=ju+ap
4409 IF INKEY$="S" THEN  GO TO 5
500
4412 IF INKEY$="N" AND env=1 THE
N  LET ju=0: LET piesp=piesp+1:
GO SUB w: LET egosp=1: GO TO 499
0
4415 IF INKEY$="N" AND env>1 THE
N  LET piesp=piesp+ju-ap: GO SUB
 w: LET ju=0: LET egosp=1: GO TO
 4990
```

```
4418 IF INKEY$="O" THEN  GO TO 4
300
4419 IF INKEY$="E" THEN  LET env
=env+1: GO TO 4424
4421 IF INKEY$<>"S" OR INKEY$<>"
N" OR INKEY$<>"O" OR INKEY$<>"E"
 THEN  GO TO 4403
4424 PRINT AT 11,0;"
 CUANTO?                 ": INPUT z
4427 IF z<2 THEN  GO TO 4424
4430 IF z<INT (RND*10)+2 THEN  G
O TO 4442
4433 PRINT AT 11,0;" ES DEMASIAD
O PARA MI, NO QUIERO": PAUSE t:
LET pie=pie+ju: GO SUB w: LET ju
=0: LET ego=1: GO TO 4990
4442 PRINT AT 11,0;"           !
!QUIERO!!       ": PAUSE t:
LET ju=ju+z: GO TO 5500
4450 REM ORDAGO DE SPECTRUM
4453 PRINT AT 11,0;"      ORDAGO
!! QUIERES? S/N      ": PAUSE 0
4456 IF INKEY$="N" AND env=1 THE
N  LET piesp=piesp+ju: GO SUB w:
 LET ju=0: LET egosp=1: GO TO 49
90
4459 IF INKEY$="N" THEN  LET pie
sp=piesp+1: GO SUB w: LET ju=0:
LET egosp=1: GO TO 4990
4462 IF INKEY$="S" THEN  GO TO 4
468
4465 IF INKEY$<>"S" OR INKEY$<>"
N" THEN GO TO 4453
4468 GO TO 4353
4800 PRINT AT 11,0;"          YO
TENGO JUEGO          ": PAUSE t:
GO TO 5500
4900 PRINT AT 11,0;"       YO NO
 TENGO JUEGO!     ": PAUSE t:
GO TO 5500
4950 PRINT AT 11,0;"       YO NO
 TENGO JUEGO!        ": LET noju=
1: PAUSE t: GO TO 5000
4990 IF totsp>=25 THEN  GO TO 65
00
4993 IF tot>=25 THEN  GO TO 6600
4996 GO TO 5500
5000 PRINT AT 11,0;"         JUG
ADA A PUNTO           ": PAUSE t
5003 LET pu=0: LET env=0: DIM s(
5): LET s=5: LET nto=0: LET ntos
p=0
5006 IF cantsp=30 THEN  LET s(s)
=1
5009 IF cantsp=29 THEN  LET s(s)
=2
5012 IF cantsp=28 THEN  LET s(s)
=3
5015 IF cantsp=27 THEN  LET s(s)
=4
5018 IF cantsp=26 THEN  LET s(s)
=5
5020 IF sp=1 THEN  GO TO 5200
5030 PRINT AT 11,0;"     ENVIDAS
A PUNTO? S/N/O/      ": PAUSE 0
5033 IF INKEY$="N" THEN  LET no=
1: GO TO 5200
```

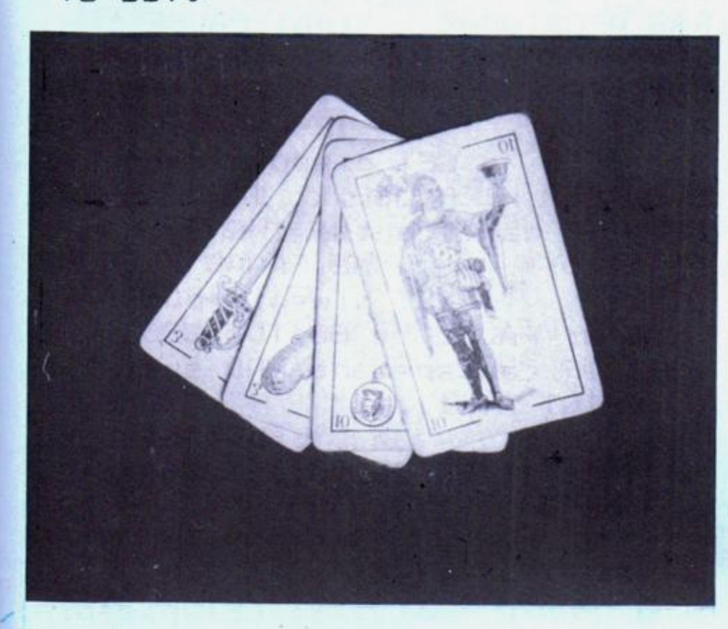


```
5036 IF INKEY$="O" THEN  GO TO 5
300
5039 IF INKEY$="S" THEN  GO TO 5
045
5042 IF INKEY$<>"N" OR INKEY$<>"
O" OR INKEY$<>"S" THEN GO TO 50
30
5045 PRINT AT 11,0;"
 CUANTO?            ": INPUT z
5048 IF z<2 THEN  GO TO 5045
5050 LET env=env+1
5051 LET pu=pu+z
5054 IF s(s)=1 AND RND>.5 THEN
GO TO 5450
5057 IF s(s)=1 THEN  LET ap=INT
(RND*4)+2: GO TO 5400
5060 IF s(s)=2 THEN  LET ap=INT
(RND*2)+2: GO TO 5400
5063 IF s(s)=3 THEN  LET ap=2: G
O TO 5400
5066 IF RND<.1 THEN  LET ap=2: G
O TO 5400
```

```
5069 IF RND>.3 THEN GO TO 5150
5100 PRINT AT 11,0;"
 QUIERO        ": PAUSE t:
GO TO 5500
5150 PRINT AT 11,0;"         N
O QUIERO!      ": PAUSE t
5151 LET nto=1
5153 IF env=1 THEN LET pu=0: LE
T pie=pie+1: GO SUB w: GO TO 549
0
5200 REM JUGADOR NO ENVIDA
5203 IF s(s)=1 AND RND>.5 THEN
GO TO 5450
5206 IF s(s)=1 THEN LET ap=INT
(RND*4)+2: GO TO 5400
5209 IF s(s)=2 AND RND>.75 THEN
 GO TO 5450
5212 IF s(s)=2 THEN LET ap=INT
(RND*2)+2: GO TO 5400
5215 IF s(s)=3 AND RND>.9 THEN
GO TO 5450
5218 IF s(s)=3 THEN LET ap=2: G
O TO 5400
5221 IF RND<.3 THEN LET ap=2: G
O TO 5400
5239 PRINT AT 11,0;"          YO
PASO A PUNTO     ": PAUSE t
5242 IF sp=1 AND no=1 THEN LET
no=0: GO TO 5500
5243 IF sp=1 THEN GO TO 5030
5245 GO TO 5490
5300 REM ORDAGO
5303 PRINT AT 11,0;"          ME
 ORDAGUEAS! EH?    ": PAUSE t
5306 IF s(s)=1 AND RND>.5 THEN
GO TO 5350
5309 IF s(s)=2 AND RND>.75 THEN
 GO TO 5350
5312 IF s(s)=3 AND RND>.9 THEN
GO TO 5350
5318 PRINT AT 11,0;"
NO QUIERO!      ": PAUSE t
5319 LET nto=1
5320 IF env=0 THEN LET pie=pie+
1: GO SUB w: LET pu=0: GO TO 549
0
5321 IF env>=1 THEN LET pie=pie
+pu: GO SUB w: LET pu=0: GO TO 5
490
5350 PRINT AT 11,0;"
!!QUIERO!!        ": PAUSE t
5353 PRINT AT 11,0;"     VAMOS A
 VER LAS CARTAS    ": PAUSE t:
GO SUB 6000
5356 IF cantsp>cant THEN GO TO
5370
5359 PRINT AT 11,0;"          ME
HAS GANADO!        ": PAUSE t:
GO TO 7000
5370 PRINT AT 11,0;"        !!TE
 HE GANADO!!      ": PAUSE t:
GO TO 7000
5400 REM SPECTRUM ENVIDA
5402 LET env=env+1
5403 PRINT AT 11,0;"YO TE ENVIDO
 ";ap;" QUIERES S/N/E/O": PAUSE
0
5406 LET pu=pu+ap
5409 IF INKEY$="S" THEN GO TO 5
500
5412 IF INKEY$="N" AND env=1 THE
N LET pu=0: LET piesp=piesp+1:
GO SUB w: LET ntosp=1: GO TO 549
0
5415 IF INKEY$="N" AND env>1 THE
N LET piesp=piesp+pu-ap: GO SUB
 w: LET pu=0: LET ntosp=1: GO TO
 5490
5418 IF INKEY$="O" THEN GO TO 5
300
5419 IF INKEY$="E" THEN LET env
=env+1: GO TO 5424
5421 IF INKEY$<>"S" OR INKEY$<>"
N" OR INKEY$<>"O" OR INKEY$<>"E"
 THEN GO TO 5403
5424 PRINT AT 11,0;"
 CUANTO?           ": INPUT z
5427 IF z<2 THEN GO TO 5424
5430 IF z<INT (RND*10)+2 THEN G
O TO 5442
5433 PRINT AT 11,0;" ES DEMASIAD
O PARA MI, NO QUIERO": PAUSE t:
LET pie=pie+pu: GO SUB w: LET ju
=0: LET nto=1: GO TO 5490
5442 PRINT AT 11,0;"           !
!QUIERO!!          ": PAUSE t:
LET pu=pu+z: GO TO 5500
5450 REM ORDAGO DE SPECTRUM
5453 PRINT AT 11,0;"      ORDAGO
!! QUIERES? S/N      ": PAUSE 0
5456 IF INKEY$="N" AND env=1 THE
N LET piesp=piesp+pu: GO SUB w:
 LET pu=0: LET ntosp=1: GO TO 54
90
5459 IF INKEY$="N" THEN LET pie
sp=piesp+1: GO SUB w: LET pu=0:
```

```
LET ntosp=1: GO TO 5490
5462 IF INKEY$="S" THEN  GO TO 5
468
5465 IF INKEY$<>"S" OR INKEY$<>"
N" THEN  GO TO 5453
5468 GO TO 5353
5490 IF totsp>=25 THEN  GO TO 65
00
5495 IF tot>=25 THEN  GO TO 6600
5500 PRINT AT 11,0;"          VAM
OS A CONTAR          ": PAUSE t
5503 PRINT AT 11,0;"       VAMOS A
 VER LAS CARTAS      ": GO SUB 60
00: PAUSE t
5504 IF an=1 THEN  PRINT AT 11,0
;"      HEMOS PASADO A GRANDE
  ": PAUSE t: GO TO 5518
5506 PRINT AT 11,0;" GRANDE, PIE
DRAS APOSTADAS  ";gr: PAUSE t
5509 IF gr=0 THEN  GO TO 5550
5512 IF gransp>gran THEN  PRINT
AT 11,0;"          TE LAS GANO!
       ": LET piesp=piesp+gr:
GO SUB w: PAUSE t: GO TO 5520
5515 IF gransp<=gran THEN  PRINT
 AT 11,0;"          ME LAS GANAS
         ": LET pie=pie+gr: GO
SUB w: PAUSE t: GO TO 5520
5518 IF gransp>gran THEN  PRINT
AT 11,0;"     TENGO LA GRANDE MA
YOR!  ": LET piesp=piesp+1: P
AUSE t: GO SUB w: GO TO 5520
5519 IF gran>gransp THEN  PRINT
AT 11,0;"     TIENES LA GRANDE M
AYOR    ": LET pie=pie+1: PAUSE
 t: GO SUB w: GO TO 5520
5520 IF totsp>=25 THEN  GO TO 65
00
5530 IF tot>=25 THEN  GO TO 6600
5550 IF eq=1 THEN  PRINT AT 11,0
;"      HEMOS PASADO A PEQUEÑA
  ": PAUSE t: GO TO 5565
5551 PRINT AT 11,0;" PEQUEÑA, PI
EDRAS APOSTADAS ";pe: PAUSE t
5553 IF pe=0 THEN  GO TO 5600
5556 IF peqsp>peq THEN  PRINT AT
 11,0;"          TE LAS GANO!
     ": LET piesp=piesp+pe: GO
 SUB w: PAUSE t: GO TO 5570
5559 IF peq>peqsp THEN  PRINT AT
 11,0;"          ME LAS GANAS
     ": LET pie=pie+pe: GO SUB
 w: PAUSE t: GO TO 5570
5565 IF peqsp>peq THEN  PRINT AT
 11,0;"      TENGO LA PEQUEÑA MEN
OR      ": LET piesp=piesp+1: PAU
SE t: GO SUB w: GO TO 5570
5568 IF peq>peqsp THEN  PRINT AT
 11,0;"      TIENES LA PEQUEÑA ME
NOR     ": LET pie=pie+1: PAUSE t
: GO SUB w: GO TO 5570
5570 IF totsp>=25 THEN  GO TO 65
00
5573 IF tot>=25 THEN  GO TO 6600
5600 IF nopa=1 THEN  PRINT AT 11
,0;"          NO HAY PARES
    ": PAUSE t: GO TO 5700
5604 IF res=1 AND o(o)=3 THEN  P
RINT AT 11,0;"          LLEVAS DU
PLES!        ": LET pie=pie+3:
GO SUB w: PAUSE t: GO TO 5690
5606 IF res=1 AND o(o)=2 THEN  P
RINT AT 11,0;"          LLEVAS ME
DIAS!        ": LET pie=pie+2:
GO SUB w: PAUSE t: GO TO 5690
5608 IF res=1 AND o(o)=1 THEN  P
RINT AT 11,0;"          LLEVAS PA
RES!!        ": LET pie=pie+1:
GO SUB w: PAUSE t: GO TO 5690
5610 IF resp=1 AND q(q)=3 THEN
PRINT AT 11,0;"           LLEVO D
UPLES         ": LET piesp=pies
p+3: GO SUB w: PAUSE t: GO TO 56
90
5612 IF resp=1 AND q(q)=2 THEN
PRINT AT 11,0;"           LLEVO M
EDIAS         ": LET piesp=pies
p+2: GO SUB w: PAUSE t: GO TO 56
90
5614 IF resp=1 AND q(q)=1 THEN
PRINT AT 11,0;"           LLEVO P
ARES          ": LET piesp=pies
p+1: GO SUB w: PAUSE t: GO TO 56
90
5616 PRINT AT 11,0;" PARES, PIED
RAS APOSTADAS   ";pa: PAUSE 150
5618 IF pa=0 THEN  GO TO 5650
5620 IF q(q)=3 AND o(o)=3 AND gr
ansp>gran THEN  GO SUB 5680: LET
 piesp=piesp+pa+3: GO SUB w: PAU
SE t: GO TO 5690
5621 IF q(q)=3 AND o(o)=3 AND gr
ansp<gran THEN  GO SUB 5685: LET
 pie=pie+pa+3: GO SUB w: PAUSE t
: GO TO 5690
5622 IF q(q)=3 AND (o(o)=2 OR o(
```

```
o)=1) THEN  GO SUB 5680: LET pie
sp=piesp+pa+3: GO SUB w: PAUSE t
: GO TO 5690
5623 IF o(o)=3 AND (q(q)=2 OR q(
q)=1) THEN  GO SUB 5685: LET pie
=pie+pa+3: GO SUB w: PAUSE t: GO
 TO 5690
5624 IF q(q)=2 AND o(o)=2 AND er
sp>er THEN  GO SUB 5680: LET pie
sp=piesp+pa+2: GO SUB w: PAUSE t
: GO TO 5690
5625 IF q(q)=2 AND o(o)=2 AND er
>ersp THEN  GO SUB 5685: LET pie
=pie+pa+2: GO SUB w: PAUSE t: GO
 TO 5690
```



```
5626 IF q(q)=2 AND o(o)=1 THEN
GO SUB 5680: LET piesp=piesp+pa+
2: GO SUB w: PAUSE t: GO TO 5690
5627 IF o(o)=2 AND q(q)=1 THEN
GO SUB 5685: LET pie=pie+pa+2: G
O SUB w: PAUSE t: GO TO 5690
5628 IF q(q)=1 AND o(o)=1 AND er
sp>er THEN  GO SUB 5680: LET pie
sp=piesp+pa+1: GO SUB w: PAUSE t
: GO TO 5690
5629 IF q(q)=1 AND o(o)=1 AND er
>ersp THEN  GO SUB 5685: LET pie
=pie+pa+1: GO SUB w: PAUSE t: GO
 TO 5690
5630 IF q(q)=1 AND o(o)=1 AND er
sp=er AND sp=1 THEN  GO SUB 5680
: LET piesp=piesp+pa+1: GO SUB w
: PAUSE t: GO TO 5690
5631 IF q(q)=1 AND o(o)=1 AND er
sp=er AND sp=0 THEN  GO SUB 5685
: LET pie=pie+pa+1: GO SUB w: PA
USE t: GO TO 5690
5650 IF q(q)=1 AND o(o)=1 AND er
sp>er THEN  GO SUB 5670: LET pie
sp=piesp+1: GO SUB w: PAUSE t: G
O TO 5690
5652 IF q(q)=1 AND o(o)=1 AND er
>ersp THEN  GO SUB 5675: LET pie
=pie+1: GO SUB w: PAUSE t: GO TO
 5690
5654 IF q(q)=1 AND o(o)=1 AND er
sp=er AND sp=1 THEN  GO SUB 5670
: LET piesp=piesp+1: GO SUB w: P
AUSE t: GO TO 5690
5656 IF q(q)=1 AND o(o)=1 AND er
sp=er AND sp=0 THEN  GO SUB 5675
: LET pie=pie+1: GO SUB w: PAUSE
 t: GO TO 5690
5670 PRINT AT 11,0;"       LOS P
ARES SON MIOS        ": RETURN
5675 PRINT AT 11,0;"       LOS PA
RES SON TUYOS!       ": RETURN
5680 PRINT AT 11,0;"        TE GA
NO LOS PARES!        ": RETURN
5685 PRINT AT 11,0;"        ME GA
NAS LOS PARES        ": RETURN
5690 IF totsp>=25 THEN  GO TO 65
00
5695 IF tot>=25 THEN  GO TO 6600
5700 IF noju=1 THEN  PRINT AT 11
,0;"          NO HAY JUEGO
   ": PAUSE t: GO TO 5800
5703 IF egosp=1 AND cantsp=31 TH
EN  PRINT AT 11,0;"           TE
NGO 31               ": LET piesp=
piesp+3: GO SUB w: PAUSE t: GO T
O 5790
5706 IF ego=1 AND cant=31 THEN
PRINT AT 11,0;"           TIENES
 31              ": LET pie=pie+3:
 GO SUB w: PAUSE t: GO TO 5790
5709 IF egosp=1 AND cantsp>31 TH
EN  PRINT AT 11,0;"           TE
NGO JUEGO            ": LET piesp=
piesp+2: GO SUB w: PAUSE t: GO T
O 5790
5712 IF ego=1 AND cant>31 THEN
PRINT AT 11,0;"           TIENES
 JUEGO           ": LET pie=pie+2:
 GO SUB w: PAUSE t: GO TO 5790
5715 PRINT AT 11,0;"JUEGO, PIEDR
AS APOSTADAS     ";ju: PAUSE t
5718 IF ju=0 THEN  GO TO 5750
```

```
5721 IF cantsp=31 AND cant<>31 T
HEN  GO SUB 5740: LET piesp=pies
p+ju+3: GO SUB w: PAUSE t: GO TO
 5790
5722 IF cant=31 AND cantsp<>31 T
HEN  GO SUB 5745: LET pie=pie+ju
+3: GO SUB w: PAUSE t: GO TO 579
0
5723 IF cantsp=32 AND cant=31 AN
D sp=1 THEN  GO SUB 5740: LET pi
esp=piesp+ju+3: GO SUB w: PAUSE
t: GO TO 5790
5724 IF cantsp=31 AND cant=31 AN
D sp=0 THEN  GO SUB 5745: LET pi
e=pie+ju+3: GO SUB w: PAUSE t: G
O TO 5790
5725 IF cantsp=32 AND (cant<>31
OR cant<>32) THEN  GO SUB 5740:
LET piesp=piesp+ju+2: GO SUB w:
PAUSE t: GO TO 5790
5726 IF cant=32 AND (cantsp<>31
OR cantsp<>32) THEN  GO SUB 5745
: LET pie=pie+ju+2: GO SUB w: PA
USE t: GO TO 5790
5727 IF cantsp=cant AND sp=1 THE
N  GO SUB 5740: LET piesp=piesp+
ju+2: GO SUB w: PAUSE t: GO TO 5
790
5728 IF cantsp=cant AND sp=0 THE
N  GO SUB 5745: LET pie=pie+ju+2
: GO SUB w: PAUSE t: GO TO 5790
5729 IF (cantsp<>31 OR cantsp<>3
2) AND (cant<>31 OR cant<>32) AN
D cantsp>cant THEN  GO SUB 5740:
 LET piesp=piesp+ju+2: GO SUB w:
 PAUSE t: GO TO 5790
5730 IF (cantsp<>31 OR cantsp<>3
2) AND (cant<>31 OR cant<>32) AN
D cant>cantsp THEN  GO SUB 5745:
 LET pie=pie+ju+2: GO SUB w: PAU
SE t: GO TO 5790
5740 PRINT AT 11,0;"         TE G
ANO EL JUEGO        ": RETURN
5745 PRINT AT 11,0;"         ME G
ANAS EL JUEGO        ": RETURN
5750 IF cantsp>cant THEN  GO SUB
 5780: LET piesp=piesp+2: GO SUB
 w: PAUSE t: GO TO 5790
5753 IF cant>cantsp THEN  GO SUB
 5785: LET pie=pie+2: GO SUB w:
PAUSE t: GO TO 5790
5756 IF cantsp=cant AND sp=1 THE
N  GO SUB 5780: LET piesp=piesp+
2: GO SUB w: PAUSE t: GO TO 5790
5757 IF cantsp=cant AND sp=0 THE
N  GO SUB 5785: LET pie=pie+2: G
O SUB w: PAUSE t: GO TO 5790
5780 PRINT AT 11,0;"         EL J
UEGO ES MIO          ": RETURN
5785 PRINT AT 11,0;"        EL JU
EGO ES TUYO          ": RETURN
5790 IF totsp>=25 THEN  GO TO 65
00
5795 IF tot>=25 THEN  GO TO 6600
5797 GO TO 5900
5800 IF nto=1 THEN  GO TO 5820
5803 IF ntosp=1 THEN  GO TO 5825
5806 PRINT AT 11,0;" PUNTO, PIED
RAS APOSTADAS    ";pu: PAUSE t
5809 IF pu=0 THEN  GO TO 5850
5812 IF cantsp>cant THEN  PRINT
AT 11,0;"         TE GANO EL PUNT
O         ": LET piesp=piesp+pu+1
: GO SUB w: PAUSE t: GO TO 5890
5815 IF cant>cantsp THEN  PRINT
AT 11,0;"        ME GANAS EL PUNT
O         ": LET pie=pie+pu+1: GO
 SUB w: PAUSE t: GO TO 5890
5816 IF cantsp=cant AND sp=1 THE
N  PRINT AT 11,0;"        TE GANO
 EL PUNTO          ": LET piesp=p
iesp+pu+1: GO SUB w: PAUSE t: GO
 TO 5890
5817 IF cantsp=cant AND sp=0 THE
N  PRINT AT 11,0;"        ME GANA
S EL PUNTO          ": LET pie=pie
+pu+1: GO SUB w: PAUSE t: GO TO
5890
5820 PRINT AT 11,0;"          EL P
UNTO ES TUYO         ": LET pie=p
ie+1: GO SUB w: PAUSE t: GO TO 5
890
5825 PRINT AT 11,0;"          EL P
UNTO ES MIO!         ": LET piesp
=piesp+1: GO SUB w: PAUSE t: GO
TO 5890
5850 IF cantsp>cant THEN  PRINT
AT 11,0;"          EL PUNTO ES MIO
          ": LET piesp=piesp+1: G
O SUB w: PAUSE t: GO TO 5890
5853 IF cant>cantsp THEN  PRINT
AT 11,0;"         EL PUNTO ES TUYO
          ": LET pie=pie+1: GO SU
B w: PAUSE t: GO TO 5890
5856 IF cantsp=cant AND sp=1 THE
N  PRINT AT 11,0;"          EL PUN
```

```
TO ES MIÓ               ": LET piesp=p
iesp+1: GO SUB w: PAUSE t: GO TO
 5890
5859 IF cantsp=cant AND sp=0 THE
N  PRINT AT 11,0;"          EL PUN
TO ES TUYO          ": LET pie=pie
+1: GO SUB w: PAUSE t: GO TO 589
0
5890 IF totsp>=25 THEN  GO TO 65
00
5895 IF tot>=25 THEN  GO TO 6600
5900 PRINT AT 11,0;"    VAMOS A
DAR NUEVAS CARTAS    "
5910 FOR n=0 TO 6
5913 PRINT AT n,8; PAPER 4;"
              "
5919 NEXT n
5922 FOR n=15 TO 21
5925 PRINT AT n,8; PAPER 4;"
              "
5931 NEXT n
5934 LET sp=sp+1
5937 IF sp>1 THEN  LET sp=0
5940 GO TO 150
6000 REM RUTINA APARICION
        CARTAS SPECTRUM
6003 LET l=0
6009 FOR n=5 TO 8
6012 FOR m=1 TO 5
6015 PRINT  PAPER 7;AT m,9+l;"
 "
6018 NEXT m
6021 PLOT 71+8*l,168: DRAW 25,0:
 DRAW 0,-41: DRAW -25,0: DRAW 0,
41
6024 IF d$(n)="♈" THEN  LET ink=
4
6027 IF d$(n)="↓" THEN  LET ink=
1
6030 IF d$(n)="●" THEN  LET ink=
6
6033 IF d$(n)="⟍" THEN  LET ink=
2
6036 PRINT  PAPER 7; INK 0;AT 1,
9+l;c$(n);AT 2,9+l; INK ink;d$(n
)
6039 PRINT  PAPER 7; INK ink;AT
4,11+l;d$(n);AT 5,11+l; INK 0;c$
(n)
6042 LET l=l+4
6043 BEEP .1,40
6045 NEXT n
6050 RETURN
```

```
6500 PRINT  FLASH 1;AT 11,0;"
 TE HE GANADO LA PARTIDA!      "
6503 PAUSE t
6506 GO TO 7000
6600 PRINT AT 11,0;"    ME HAS G
ANADO LA PARTIDA     "
6603 PAUSE t
6606 GO TO 7000
7000 CLS : PRINT AT 11,0;"   JUG
AMOS OTRA PARTIDA?  S/N    ": PAU
SE 0
7003 IF INKEY$="S" THEN  GO TO 1
00
7006 IF INKEY$="N" THEN  PRINT A
T 11,0;"            !!ADIOS!!!
         ": PAUSE t: RANDOMIZE US
R 0
7009 IF INKEY$<>"S" OR INKEY$<>"
N" THEN  GO TO 7000
```

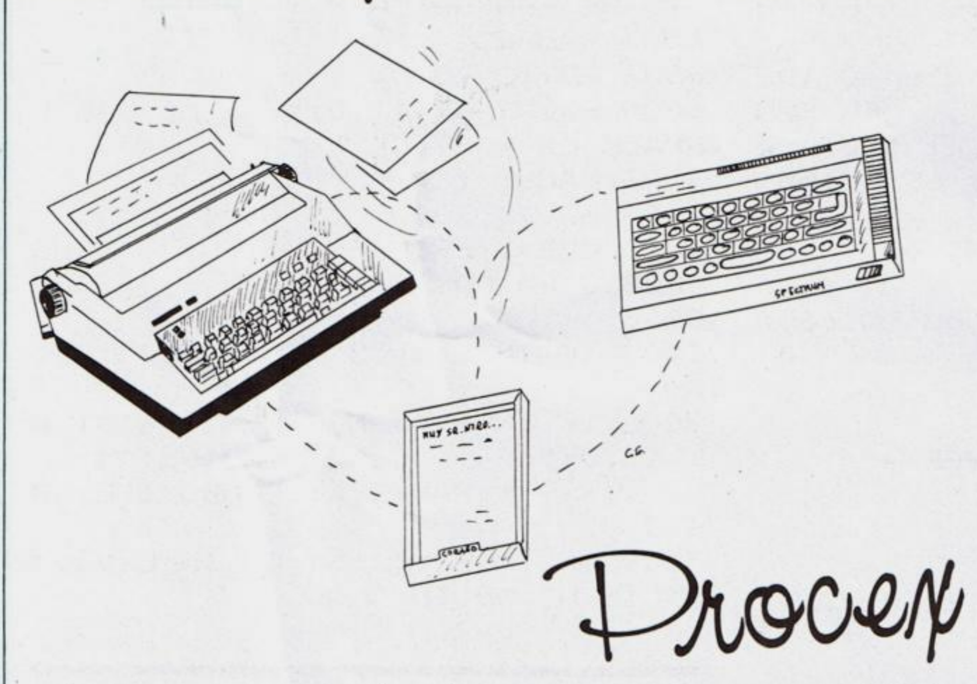
FIGURA 2

Tabla de caracteres gráficos

```
♈ ↓ ● ⟍ ⌧ 10 11 12 I J K L M Ñ O P Q R S T U

A B C D E F G H I J K L M N O P Q R S T U
```

```
   5 REM CARGADOR DE FIGURAS
  10 FOR n=65368 TO 65431
  20 READ a
  30 POKE n,a
  40 NEXT n
  50 SAVE "figuras"CODE 65368,64
  60 DATA 0,124,124,56,16,16,56,
0
  70 DATA 0,16,16,16,16,56,16,16
  80 DATA 0,24,60,60,60,60,24,0
  90 DATA 0,96,112,56,28,12,6,2
 100 DATA 129,90,36,165,165,36,9
0,129
 110 DATA 0,46,106,42,42,42,46,0
 120 DATA 0,36,108,36,36,36,36,0
 130 DATA 0,46,106,34,36,40,46,0
```



Procex

Este programa nos permite disponer de un sencillo procesador de textos de uso fácil pero con características muy potentes.

Una de las característica más interesantes es que está escrito en BASIC usando la ampliación del sistema operativo publicada en el número 13 de esta misma revista. Esto permite usar 64 columnas conservando la estructura del BASIC, lo que permite modificar y ampliar el programa.

Su uso es sencillo. Al cargar el programa (habiéndolo tecleado y salvado previamente con SAVE «PROCEX" LINE 1) aparece un menú principal de ocho opciones. La primera entra en el editor, la segunda nos saca el texto por impresora, la tercera borra el texto del editor, la cuarta y quinta graban y leen el texto de cassette, la sexta es un pequeño menú de ayuda, la séptima cambia los colores para que sean más agradables a la vista y por último la octava termina el programa y nos devuelve al BASIC.

Dentro del editor existen cuatro teclas de función (por desgracia las flechas de movimiento no las toma en consideración). EDIT vuelve al menú principal, CAPS LOCK pone en mayúsculas, TRUE VIDEO pasa a minúsculas y DELETE, como es usual, borra el caracter habitual al que se encuentra el cursor.

La salida por impresora está preparada para ZX Printer o GP 50, para otras habrá que realizar algunos cambios entre las líneas 420 y 520. En especial habrá que pasar a MODE 0 antes de empezar y MODE 1 al terminar, así como sustituir los PRINT por LPRINT y modificar los saltos de línea. Esto variará con cada impresora y resultará fácil en la mayoría.■

```
  10 REM  P R O C E X  1.985 PARA
 ARANA SOFTWARE
  20 LET g=0: LET ant=40000: LET
 fin=40000: LET direcc=40000: GO
 TO 50
  30 IF l>21 THEN  CLS : LET l=0
: LET c=0
  40 PRINT ^0;AT 0,0;".": PRINT
^0;AT 1,0;"LINEA:";AT 1,7;l;AT 1
,10;"COLUMNA:";AT 1,19;c;AT 1,23
;"DIRECCION:";AT 1,34;direcc;AT
1,40;"Operacion:ESCRIBIR TEXTO":
 RETURN
  50 MODE 1
  60 PAPER 1: BORDER 1: INK 7: C
LS
  70 POKE 23658,0: LET l=0: LET
c=0: LET direcc=40000
  80 PRINT AT 1,17; BRIGHT 1; FL
ASH 1;" MENU GENERAL DE OPCIONES
 "
  90 PRINT AT 3,17;"1.-  ESCRIBI
R TEXTO";AT 5,17;"2.-  IMPRESORA
 ";AT 7,17;"3.-  BORRADO DE TEXT
O";AT 9,17;"4.-  SAVE";AT 11,17;
"5.-  LOAD";AT 13,17;"6.-  INFOR
MACION";AT 15,17;"7.-  CAMBIO CO
LORES";AT 17,17;"8.-  SALIDA PRO
GRAMA"
 100 PRINT AT 21,0; INVERSE 1;"
  P R O C E X     (C) Jose Antoni
o Rodriguez 85   Ver. 1,0       "
 110 LET mem=(PEEK 23730+(256*PE
EK 23731))-(PEEK 23653+(256*PEEK
 23654)): PRINT AT 19,50;"MEMORI
A:"; BRIGHT 1;mem
 120 PRINT ^0;AT 0,0;"Operacion:
"; BRIGHT 1;"MENU GENERAL
Pulsa una opcion": PAUSE 0: LET
a$=INKEY$
 130 IF a$<"1" OR a$>"8" THEN  G
O TO 120: ELSE  LET a=VAL a$
```

59

```
 140 ON a GO TO 150,420,530,620,
710,770,820,870
 150 REM  ESCRIBIR TEXTO
 160 LET direcc=fin+1: CLS
 170 POKE 23658,0: PRINT ^0;AT 0
,0;"----------------------------
----------------------------------
--L-"
 180 GO SUB 880
 190 PRINT AT l,c;"_": BEEP .01,
0
 200 PRINT ^0;AT 1,0;"Linea:";AT
 1,7;l;AT 1,10;"Columna:";AT 1,1
9;c;AT 1,23;"DIRECCION:";AT 1,34
;direcc;AT 1,40;"Operacion:ESCRI
BIR TEXTO"
 210 PAUSE 0: LET a=CODE INKEY$:
 BEEP .01,0
 220 IF a=6 THEN  POKE 23658,8:
PRINT ^0;AT 0,62; INVERSE 1;"C"
 230 IF a=4 THEN  POKE 23658,0:
PRINT ^0;AT 0,62; INVERSE 1;"L"
 240 IF a=7 THEN  CLS : GO TO 60
 250 IF a<12 THEN  GO TO 210
 260 IF c>63 THEN  LET c=0: LET
l=l+1: GO SUB 30: GO TO 300
 270 IF c=60 THEN  BEEP .02,30
 280 IF a=13 THEN  PRINT AT l,c;
"  ": LET c=0: LET l=l+1: POKE d
irecc,13: LET direcc=direcc+1: P
RINT ^0;AT 1,7;l;" ": GO SUB 30:
 PRINT AT l,c;"_": GO TO 210
 290 IF a=12 THEN  GO TO 350
 300 PRINT AT l,c;CHR$ a;"_"
 310 POKE direcc,a: LET direcc=d
irecc+1: LET c=c+1
 320 LET fin=direcc
 330 PRINT ^0;AT 1,7;l;" ";AT 1,
19;c;" ";AT 1,34;direcc;" "
 340 GO TO 210
 350 LET c=c-1 AND c>0: PRINT AT
 l,c;"_ ";: LET direcc=direcc-1
 360 IF l<=0 THEN  LET l=0
 370 PRINT ^0;AT 1,7;l;" ";AT 1,
19;c;" ";AT 1,34;direcc;" "
 380 IF direcc<40000 THEN  LET d
irecc=40000
 390 IF c=0 AND l>0 THEN  PRINT
AT l,c;" ": LET c=63: LET l=l-1
AND l>0: GO TO 350
 400 GO TO 210
 410 PAUSE 0: STOP
 420 REM  IMPRESORA
 430 CLS : PRINT AT 0,0;
 440 LET c=0: LET l=0: FOR n=400
00 TO fin
 450 IF PEEK n=13 THEN  LET l=l+
1: LET c=0
 460 IF l>21 THEN  CLS : LET l=0
: LET c=0
 470 PRINT AT l,c;CHR$ (PEEK n):
 LET c=c+1
 480 POKE 23692,255
 490 IF c>63 THEN  LET c=0: LET
l=l+1
 500 IF l>21 THEN  COPY : CLS :
LET l=0: LET c=0
 510 NEXT n: COPY
 520 CLS : GO TO 60
 530 REM  BORRADO DEL TEXTO
 540 BORDER 2: PAPER 2: INK 7: C
LS
 550 PRINT  BRIGHT 1; FLASH 1;AT
 10,20;"BORRADO DEL TEXTO"
 560 INPUT "Confirmacion de la o
rden   (Si/No)  ",,,,a$
 570 IF a$="NO" OR a$="no" OR a$
="No" OR a$="n" THEN  CLS : GO T
O 60
 580 IF a$="Si" OR a$="si" OR a$
="SI" OR a$="s" THEN  FOR n=4000
0 TO 41000: POKE n,0: PRINT AT 2
1,0;41000-n;"  ": NEXT n: LET di
recc=40000
 590 LET fin=40000
 600 BORDER 7: PAPER 7: INK 0: C
LS : PRINT AT 10,20; BRIGHT 1; F
LASH 1;"BORRADO COMPLETO"
 610 BEEP 2,0: PAUSE 100: BORDER
 1: PAPER 1: INK 7: CLS : GO TO
60
 620 REM  SAVE a cassette
 630 BORDER 5: PAPER 5: INK 0: C
LS
 640 INPUT "Nombre para SAVE:",,
,,,,a$
 650 PRINT AT 10,20; BRIGHT 1; F
LASH 1;"GRABACION TEXTO";AT 12,2
0;"PULSA UNA TECLA"
 660 PRINT AT 14,20;"Tome nota:"
```

Programas (59)

```
;AT 16,20;"LONGITUD:";(fin-40000
): BEEP 3,0
 670 SAVE a$CODE 40000,(fin-4000
0)
 680 CLS : PRINT AT 10,20; BRIGH
T 1; FLASH 1;"GRABACION EFECTUAD
A";AT 12,20;"QUIERES EFECTUAR VE
RIFY": PAUSE 0
 690 IF INKEY$="s" THEN  VERIFY
a$CODE 40000,(fin-40000): CLS :
BORDER 1: PAPER 1: INK 7: PRINT
AT 10,20; BRIGHT 1; FLASH 1;"VER
IFY CORRECTO": BEEP 1,30: CLS :
GO TO 60
 700 BORDER 1: PAPER 1: INK 7: C
LS : GO TO 60
 710 REM  LOAD desde cassette
 720 BORDER 1: PAPER 1: INK 7: C
LS
 730 INPUT "LONGITUD:";long
 740 PRINT AT 10,20; BRIGHT 1; F
LASH 1;"INTRODUCE LA CINTA";AT 1
2,20;"Y PULSA PLAY.": LOAD ""COD
E 40000,long
 750 LET fin=40000+long
 760 CLS : GO TO 60
 770 REM  INFORMACION
 780 CLS : PRINT AT 0,0;"Informa
cion:"''"Las funciones de las t
eclas en el editor son las sigui
entes:"''"       CAPS SHIFT + 1 (ed
it)............... Vuelve al MEN
U               CAPS SHIFT + 2 (ca
ps lock).......... MAYUSCULAS
                CAPS SHIFT + 3 (tr
ue video)......... Minusculas
                CAPS SHIFT + 0 (de
lete)............. Borrado
                ENTER.............
.................... NUEVA LINEA"
 790 PRINT ''"          OPCIONES DE
L EDITOR"''"1.- Editor 64 ctes"'
"2.- Impresion "'"3.- Borrado de
l texto"'"4.- Grabacion texto"'"
5.- Carga del texto"'"6.- Inform
acion"'"7.- Cambio de los colore
s (0 a 7)"'"8.- SALIDA del progr
ama"
 800 PRINT AT 21,0; INVERSE 1; B
RIGHT 1;"    Pulsa una tecla para
 volver al MENU GENERAL o C para
 COPY      ": PAUSE 0: IF INKEY$="
c" OR INKEY$="C" THEN  COPY : CL
S : GO TO 60
 810 CLS : GO TO 60
 820 REM  CAMBIO DEL PAPER E INK

 830 CLS : FOR n=0 TO 7: PRINT A
T n+5,5; INK n;"ABCDEFGHIJKLMNOP
QRSTUVWXYZ";AT n+5,40; INK 7;" >
 ";n: NEXT n
 840 PRINT AT 17,10;"Color del P
APEL (paper) ": BEEP 1,2: PAUSE
0: LET a$=INKEY$: IF a$<"0" OR a
$>"7" THEN  GO TO 840: ELSE  LET
 paper=VAL a$
 850 PRINT AT 17,10;"Color de la
 tinta (INK) ": BEEP 1,0: PAUSE
0: LET a$=INKEY$: IF a$<"0" OR a
$>"7" THEN  GO TO 840: ELSE  LET
 ink=VAL a$
 860 BORDER paper: PAPER paper:
INK ink: CLS : GO TO 60
 870 CLS : BORDER 7: PAPER 7: IN
K 0: CLS : MODE 0: BEEP .2,20: S
TOP
 880 LET c=0: LET l=0: PRINT AT
0,0;: FOR n=40000 TO fin
 890 IF PEEK n=13 THEN  LET l=l+
1: LET c=0
 900 IF l>21 THEN  CLS : LET l=0
: LET c=0
 910 IF c>63 THEN  LET c=0: LET
l=l+1
 920 PRINT AT l,c;CHR$ (PEEK n):
 LET c=c+1
 930 POKE 23692,255:
 940 IF c>63 THEN  LET c=0: LET
l=l+1
 950 NEXT n
 960 RETURN
 970 ON ERR L
 980 BEEP .4,0: CLS : PRINT AT 1
0,20; BRIGHT 1; FLASH 1;">>>>>>E
RROR<<<<<<"
 990 FOR n=1 TO 100: PAUSE 2: NE
XT n: CLS : GO TO 60
1000 SAVE "PROCEX" LINE 10: VERI
FY "": STOP
```

SERVICIO DE EJEMPLARES ATRASADOS

Complete su colección de

Todospectrum

A continuación le resumimos el contenido de los ejemplares aparecidos hasta ahora.

Núm. 1 • 250 pts.
Cómo usar el microdrive/Programación Basic/Ampliación Basicare/Rutina despertador/Variables del sistema/Entrada datos mediante máscaras/Protección del software/Sintonice su Spectrum/Programas.

Núm. 2 • 250 pts.
Gráficos profesionales/Desplazamiento pixel a pixel/Utilización de rutinas/Construcción del interface centronics/Programas de utilidad para microdrive/Rutina reset en código máquina/Análisis del editor de textos Tasword/Interfaces para impresoras/Programas.

Núm. 3 • 250 pts.
Novedades sonimag '84/Ampliando el Basic/Programas para ordenar programas/Gráficos con el VU-3D/Lenguaje Forth/Archivos en microdrive/Programación de un interface de impresora/Programas.

Núm. 4 • 250 pts.
De profesión: programador/Consola para el Spectrum/Comparación código máquina-Basic/Análisis programa contabilidad /Calendario/Pascal/Programas.

Núm. 5 • 250 pts.
Floppys para Spectrum/Diseño asistido por ordenador/64 Caracteres por línea/Juego de la vida/Pascal/Así hacemos las portadas/Control de evaluaciones/Programas.

Núm. 6 • 250 pts.
Representación de funciones/Todos los caminos conducen a la ROM/Juegos/Pascal/Construcción de un lápiz óptico/Programas de gestión. El SITI/Logo: tortugas para todos/Interrupciones del Z-80/Programas.

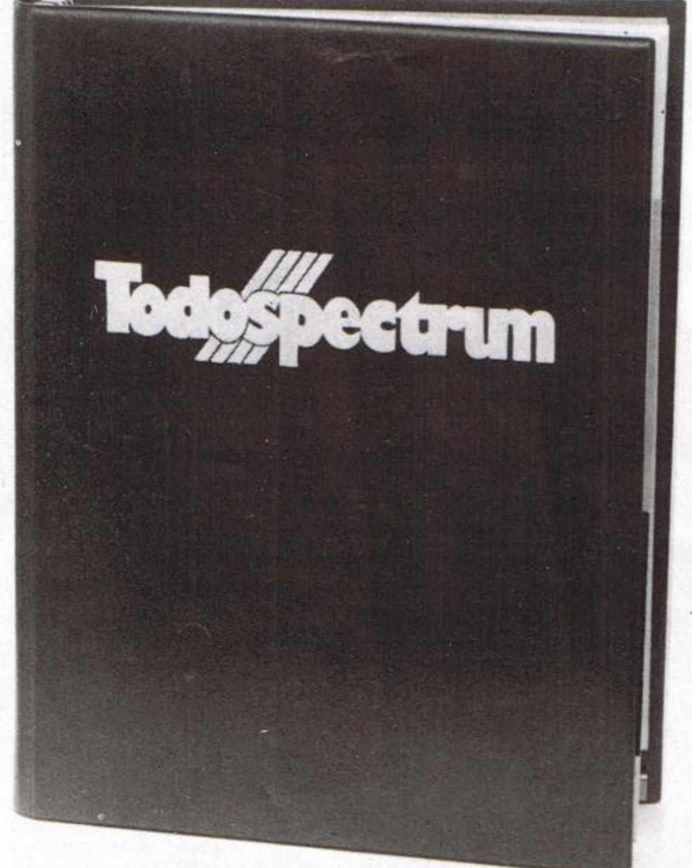




DISPONEMOS DE TAPAS ESPECIALES PARA SUS EJEMPLARES DE

Todospectrum

SIN NECESIDAD DE ENCUADERNACION



(cada tapa es para 6 ejemplares)



Para hacer su pedido, rellene este cupón HOY MISMO y envíelo a:

Todospectrum **Bravo Murillo, 377**
Tel. 733 96 62 - 28020 MADRID

Ruego me envíen los siguientes ejemplares atrasados de TODOSPECTRUM .... al precio de 250 pts.

Por favor envíenme .... tapas para la encuadernación de mis ejemplares de TODOSPECTRUM, al precio de 600 pts. más gastos de envío.

El importe lo abonaré
□ POR CHEQUE □ CONTRA REEMBOLSO □ CON MI TARJETA DE CREDITO □ AMERICAN EXPRESS □ VISA □ INTERBANK

Número de mi tarjeta: □□□□□□□□□□□□□□□□□□□□□□□□□□

Fecha de caducidad .... Firma

NOMBRE ....

DIRECCION ....

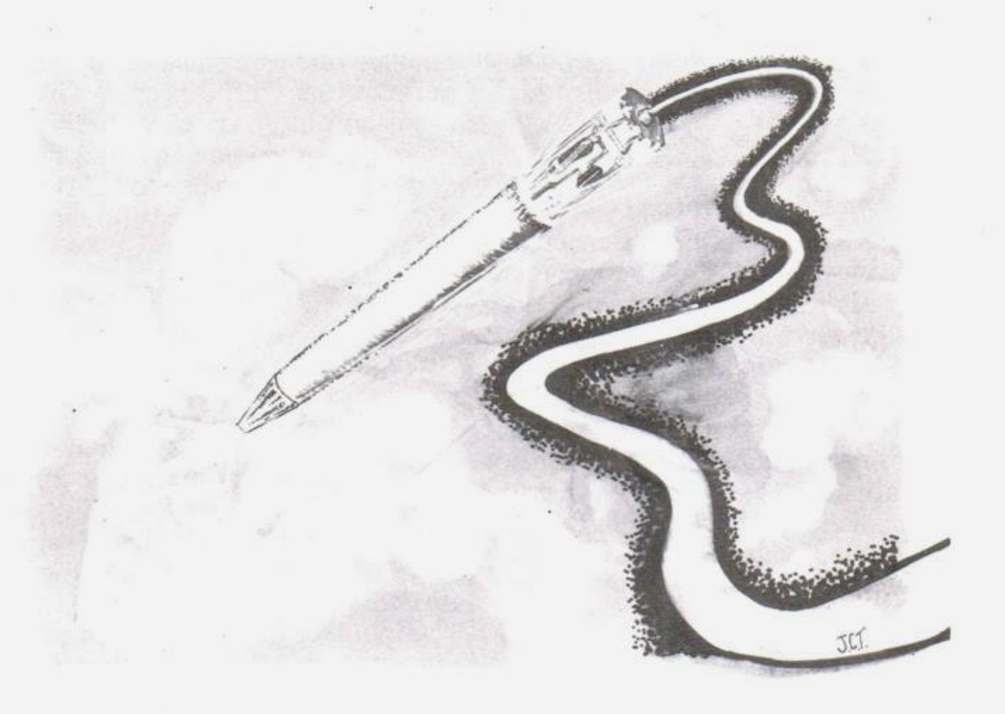
CIUDAD .... C. P. ....

PROVINCIA ....

Preguntas y respuestas

P Aunque hace tiempo que poseo mi ordenador Sinclair QL, en el número 15 de su revista, más concretamente en la sección dedicada a dicho ordenador, se citan algunas estructuras de las que ciertamente no tenía conocimiento, tales como WHEN EOF... END WHEN y WHEN ERRor... END WHEN, desearía pedirles que a ser posible me proporcionasen una descripción más detallada de dichas estructuras. Gracias anticipadas por su colaboración.

Sergio Montoro
Madrid



R *Muchos lectores han manifestado un interés similar sobre estas estructuras y sobre su modo de uso, por lo que pasaremos a describirlas brevemente, no sin antes decir que la razón de que no estén documentadas en el manual es que su funcionamiento no es del todo correcto y a veces dan problemas, por lo que su uso debe hacerse con mucho cuidado.*

La instrucción WHEN ERRor END WHEN se utiliza para detectar errores y corregirlos. Se debe situar al principio del programa de modo que el ordenador se «entere» de su existencia antes de producirse el error. Cuando se produce el error, el ordenador en lugar de pararse salta a esta rutina y después de ejecutarla salta a la instrucción siguiente a la que produjo el error. Dentro del WHEN ERR es necesario saber qué tipo de error se ha producido y dónde, de modo que podamos corregirlo. Para ello existen dos variables. ERR_LIN nos da la línea que ha producido el error y ERR_NUM da el número de este según la tabla que aparece al final del manual.

La otra estructura es WHEN EOF...END WHEN. Está destinada a ser usada al leer ficheros y hará que el programa salte a ella cuando intentemos leer de un fichero y no haya más datos en este. Al igual que en el caso anterior debe estar situada antes de la operación de lectura de ficheros.

Esperemos que con esta explicación queden aclaradas todas las dudas de estas estructuras, y recordamos, una vez más, que si no aparecen en el manual es porque no funcionan del todo bien, por lo que se debe tener mucho cuidado en su uso.

P Me gustan mucho los comentarios sobre *hardware* comercial, pero me gustaría que, además de seguir criterios de actualidad, los hicieran con un poco de orden, como ya empezaron a hacer (¿por qué no han seguido?) con los lápices ópticos, para poder comparar unos con otros. A ver cuándo publican comentarios sobre impresoras e *interfaces* centronics.

Otro tema que me parece muy interesante y del que tampoco han hablado es sobre los fallos de la ROM del Spectrum y que, a pesar de lo bien realizada que está, se les colaron una buena cantidad; por ejemplo, de los que habla el doctor Ian Logan en su libro *The complete Spectrum ROM disassembly* sobre el manejo del número 65535, o los que se producen cuando se emplean potencias (todo lo relacionado con ellas, también se producen con SQR o FN) o los fallos de SCREEN$ o VAL.

Francisco Romero
Madrid

R *Tomamos nota de todas tus sugerencias y deseos para con nuestra revista y esperamos en breve los veas cumplidos. En lo referente a los lápices ópticos, el paro momentáneo de la serie que les veníamos dedicando ha sido motivado por el simple hecho de que no disponemos de otros modelos de este periférico, y es evidente que no podemos hacer una crítica de un producto que no tenemos delante. En cuanto las casas distribuidoras correspondientes se decidan a proporcionárnoslos, prometemos continuar tratando este interesante tema.*

En cuanto a las erratas de la ROM, estamos de acuerdo contigo en que, aunque no sean muchas, es un tema que vale la pena conocer a fondo. No es esta, en todo caso, la sección apropiada para tratar un tema tan extenso, por lo que habrás de esperar a que podamos

Preguntas y respuestas

abrirle un hueco en la revista en alguno de los números siguientes.

P Felicito a esta revista y al autor del programa publicado en el n.º 13 con el nombre de «Un nuevo operativo para el Spectrum», que incluye siete nuevos comandos al BASIC de éste. Lo considero de gran calidad, y además eleva al BASIC del Spectrum al nivel de otros ordenadores de mayor precio y calidad. No obstante, al verlo, eché de menos el comando DELETE, que, como ya sabrán, sirve para borrar bloques de programa BASIC. Su sintaxis habitual suele ser:

DELETE n1,n2

siendo n1 el número de la primera línea que se desea borrar y n2 el de la última línea que ha de ser borrada.

Mis precarios conocimientos sobre código máquina me impiden la realización de este implemento, pero envío mi propuesta a su autor o a cualquier hábil programador para que se anime a incorporar este comando al ya potente programa de Manuel Arana.

Fernando Corres
Vitoria (Alava)

R *Transmitimos desde estas líneas tu petición a todos nuestros lectores y les pedimos que se animen a comunicarnos cualquier modificación o implemento de interés general, que pueda mejorar, no sólo este, sino también al resto de los programas que mes a mes venimos publicando. Caso de que algún amable lector haga eco a esta llamada, prometemos darle prioridad para que sea publicada en esta misma sección lo antes posible.*

P Me han dicho que el *Interface* 1, que poseo desde hace unos meses, además de añadir todos los comandos que especifica el manual, permite usar CLS especificando el canal en que se desea el borrado. Lo he probado con el canal 2 para la pantalla principal y con el 0 y 1 para la parte baja y no funciona. ¿A qué se debe esto?

Juan Luis Berenguer
Aguafría (Almería)

R *En efecto, el Interface* 1 incorpora no ya uno sino dos nuevos comandos que no aparecen en el manual: CLS$ y CLEAR$, pero en ellos no podrás especificar el número del canal, ya que su cometido es el siguiente:

CLS$ (sin nada detrás) borra el contenido del archivo de presentación visual y coloca los cursores de alta y baja resolución a sus respectivas esquinas (hasta aquí lo mismo que CLS), pero además coloca en las correspondientes variables del sistema y en la totalidad del archivo de atributos lo mismo que si se hubiera inicializado la máquina. Es decir pone BORDER 7, PAPER 7, INK 0, FLASH 0 e INVERSE 0. Las aplicaciones son obvias, por lo que sobran los comentarios.

CLEAR$ (también sin nada detrás) cierra todas las corrientes que hubiéramos abierto y restablece las que van del 0 al 3 a los canales asignados habitualmente (como si acabáramos de conectar el ordenador) pero respetando las variables y el archivo de pantalla. ¡Ojo!, este comando no debe utilizarse nunca como sustituto de CLOSE$, ya que los datos que se encuentren en un buffer *a medio llenar se perderían sin llegar a su destino.*

DIRECTOR:
Fernando García
COORDINADOR EDITORIAL:
Emiliano Juárez
REDACCION:
Enrique Larreta
Juan Arencibia, Fernando García, José C. Tomás, Luis M. Brugarolas, Santiago Gala
DISEÑO: Ricardo Segura

Editado por
PUBLINFORMATICA, S. A.
Presidente: Fernando Bolín

Administración:
INFODIS, S. A.
Gerente de Circulación y ventas:
Luis Carrero
Producción:
Miguel Onieva
Director de Marketing:
Antonio González
Servicio al cliente:
Julia González. Tel. 733 79 69
Administración:
Miguel Atance
Jefe de Publicidad
María José Martín
Dirección y redacción:
Bravo Murillo, 377-5.º A. Tel. 733 74 13
Telex: 48877 OPZX e 28020 Madrid
Administración y Publicidad:
Bravo Murillo, 377-3 E. Tels. 733 96 62/96
Publicidad Madrid:
María Jose Martín
Publicidad Barcelona:
María del Carmen Ríos.
Pelayo, 12.
Tel. (93) 318 02 89.
08001 Barcelona.
Depósito legal: M-29041-1984
Distribuye S.G.E.L.
Avda. Valdelaparra, s/n.
Alcobendas-Madrid.
Fotomecánica: Karmat, C/ Pantoja, 10. Madrid.
Fotocomposición: Artecomp.
Imprime: Héroes, C/ Torrelara, 8. Madrid.

Distribuidor en VENEZUELA,
SIPAM, S.A.
AVD. REPUBLICA DOMINICANA,
EDIF. FELTREC – OFICINA 4B
BOLEITA SUR
CARACAS (VENEZUELA)

Esta publicación es miembro de la Asociación de Revistas de Información ARI asociada a la Federación Internacional de Prensa Periódica, FIPP.

SUSCRIPCIONES:
Rogamos dirijan toda la correspondencia relacionada con suscripciones a:
TODOSPECTRUM
EDISA: Tel. 415 97 12
C/ López de Hoyos, 141-5.º
28002 MADRID
(Para todos los pagos reseñar solamente TODOSPECTRUM)

Para la compra de ejemplares atrasados diríjanse a la propia editorial
TODOSPECTRUM
C/ Bravo Murillo, 377-5.º A
Tel. 733 74 13 - 28020 MADRID

Si deseas colaborar en TODOSPECTRUM remite tus artículos o programas a Bravo Murillo 377, 5.º A. 28020 Madrid. Los programas deberán estar grabados en cassette y los artículos mecanografiados.
A efectos de remuneración, se analiza cada colaboración aisladamente, estudiando su complejidad y calidad.

REGALE UNA IMPRESORA A SU ORDENADOR

SEIKOSHA

GP 50	La pequeña 50 cps. Papel normal con interfaces paralelo, serial y spectrum	19.900 ptas.
SP 800 *	La perfección 96 cps. Introductor automático hoja a hoja 24 cps. en alta calidad	64.900 ptas.
SP 1000 *	La programable 100 cps. 96 cart. programables en RAM. Introductor hoja a hoja 24 cps. en alta calidad	69.900 ptas.
SP 1000 AS	La programable 100 cps. 96 cart. programables en RAM. con interface RS232	59.900 ptas.
GP 700 *	La de color 50 cps. 7 colores. 80 columnas. Tracción y fricción. Papel de 10 pulgadas	69.900 ptas.
BP 5200 *	La de oficina 200 cps. 106 cps en alta calidad. Buffer 4K. Introductor automático de documentos (Opc)	219.900 ptas.
BP 5420 *	La más rápida 400 cps. 106 cps en alta calidad. Buffer de 18K. Paralelo y RS232	319.900 ptas.

Interfaces: Serie RS232C, Spectrum, IBM, COMMODORE, MSX, QL, Apple Macintosh, HP-IB

* con interface paralelo.



Avda. Blasco Ibáñez, 116
Tel. (96) 372. 88. 89
Telex 62220 - 46022 VALENCIA

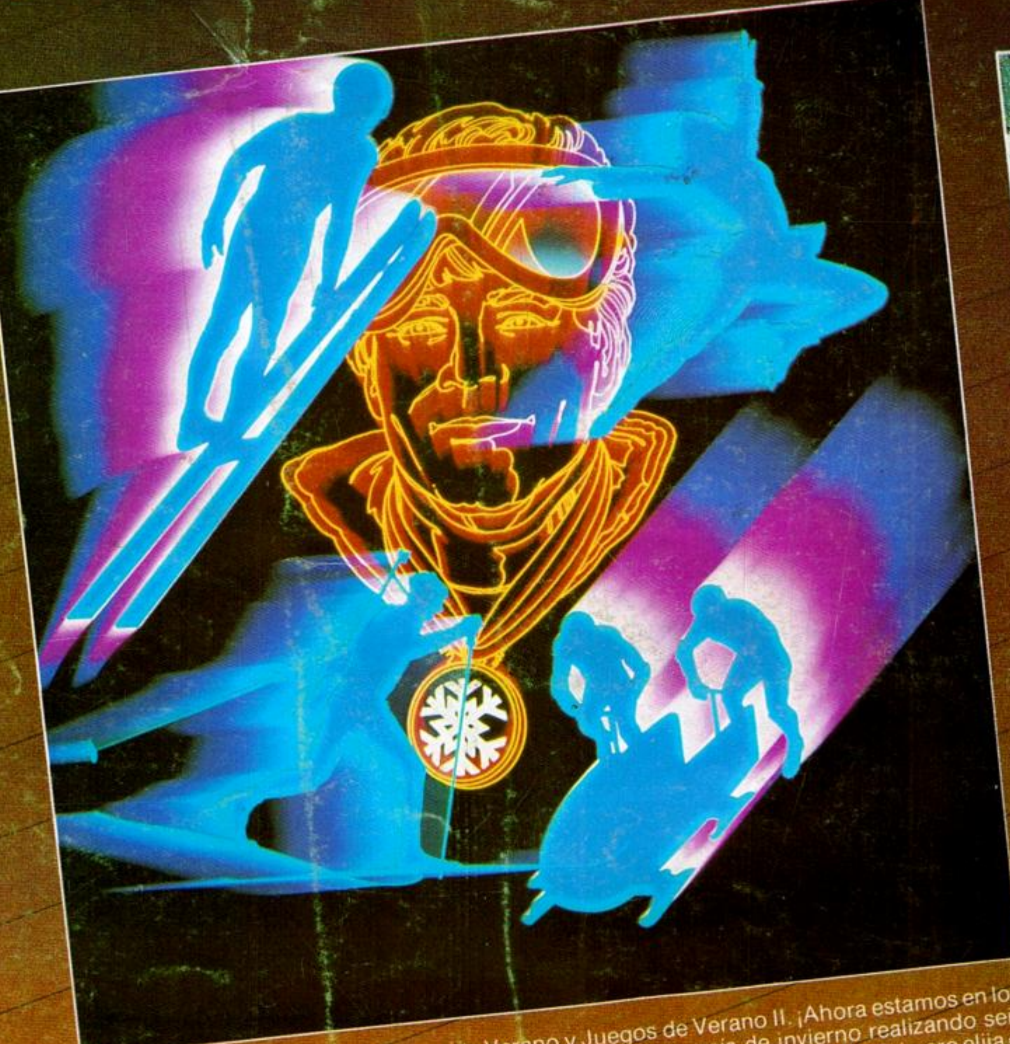
Muntaner, 60-2.º-4.ª
Tel. (93) 323. 32. 19
08011 BARCELONA

Agustin de Foxá, 25-3.º-A
Tels. (91) 733. 57. 00-733. 56. 50
28036 MADRID

Primero fué SUMMER GAMES
despues SUMMER GAMES II

y ahora...

WINTER GAMES











Ha conseguido el oro en los Juegos de Verano y Juegos de Verano II. ¡Ahora estamos en los Juegos de Invierno!, y qué increíble marco, un completo país de invierno realizando seis competiciones de acción. Puede competir contra sus amigos o el ordenador. Primero elija el país que quiera representar. Practíquelo, prepárese y aprenda una estrategia para ganar en cada competición. Ahora comience la ceremonia de apertura y la competición. ¿Será usted quien consiga el oro en la ceremonia de entrega de premios?
La búsqueda del oro continua... y está todo aquí: la estrategia, el reto, la competición, **el arte y la pompa** de los Juegos de Invierno.

- **Seis competiciones de invierno: Bobsled, salto de ski, patinaje artístico, patinaje libre estilo, Hot Dog Aéreo y el ski de fondo.**
- **Ceremonias de apertura, cierre y entrega de premios con himnos nacionales.**
- **Compita contra el ordenador o contra sus amigos o familia.**
- **Control único por el joystick, necesita destreza y cronometraje.**
- **Uno a ocho jugadores.**

EPYX
COMPUTER SOFTWARE

Fabricado y distribuido en exclusiva por:

COMPULOGICAL S.A.
Santa Cruz de Marcenado, 31 - 28015 Madrid

Distribuido en Cataluña y Baleares por: **YA ESTA DISPONIBLE PARA EL SPECTRUM**
DISCLU, S.A. - Balmes, 58 - BARCELONA - Tel. (93) 302 39 08 - **P.V.P. 2.300 Ptas.**